TEC-0010

AD-A251 277

Symbolic Model-Based SAR Feature Analysis and Change Detection

DTIC ELECTE MAY 28 1992 S A D

James C. Curlander
Wolfgang Kober

Vexcel Corporation
2477 55th Street, Suite 201
Boulder, CO 80301

February 1992

92-13942

"Original contains color plates: All DTIC reproductions will be in black and white"

Approved for public release; distribution is unlimited.

92 5 27 026

U.S. Army Corps of Engineers
Topographic Engineering Center
Fort Belvoir, Virginia 22060-5546

US Army Corps of Engineers
Topographic Engineering Center

T E C

U.S. ARMY TOPOGRAPHIC ENGINEERING CENTER

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
February 1992

3. REPORT TYPE AND DATES COVERED
Final Technical Report: May 91 - Jan 92

4. TITLE AND SUBTITLE
Symbolic Model-Based SAR Feature Analysis and Change Detection

5. FUNDING NUMBERS
DACA76-91-C-0008

6. AUTHOR(S)
James C. Curlander
Wolfgang Kober

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Vexcel Corporation
2477 55th Street, Suite 201
Boulder, CO 80301 USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Topographic Engineering Center
Fort Belvior, VA 22060-5546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TEC-0010

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The feasibility of a model-based approach to feature classification and change detection in synthetic aperture radar imagery is investigated. The models include quantitative, qualitative and relational aspects of feature signatures. The intended Phase III classification system will consist of an automated model-driven processor linked to an image understanding library such that during automated classification, the model-based processor will call various library routines as required to reach solutions. The Phase I effort consisted of:
(1) Investigation of various methodologies for rule-oriented model-based approaches;
(2) Development and test results of a numerical scoring function for tracking uncertainties during rule-based processing;
(3) Development and specification of the required image processing library functions;
(4) Development and testing of a 120 rule prototype classification system; and
(5) Results from testing the combination rule prototype and image processing library on four SAR images.
The initial testing of the 120 rule prototype over the four SAR data sets resulted in a 71% correct classification rate overall.

14. SUBJECT TERMS
model, classification, change detection, image processing, feature, SAR, automated, rule, expert system

15. NUMBER OF PAGES
115

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

PREFACE

This report describes work performed under contract DACA76-91-C-0008 for the U.S. Army Topographic Engineering Center, Fort Belvoir, Virginia 22060-5546 by Vexcel Corporation, Boulder Colorado 80301. The Contracting Officer's Technical Representative was Mr. Edmundo Simental.

1.0 Executive Overview

There is an acute need for all-weather, day and night surveillance capable of detecting and updating the locations of militarily significant targets. Modern military operations are often conducted under less than ideal environmental conditions in which conventional optical, and infrared sensors are paralyzed. Synthetic Aperture Radar (SAR) provides the needed coverage for tactical change detection. SAR sensors however, produce a wide variety of target signatures caused by variations in image scenarios. The use of symbolic models, that is models based on quantitative, qualitative and relational information, rather than simply quantitative data as procedural systems, provides the required tool for successful feature classification and change analysis of military features in SAR.

This Phase I SBIR research effort was concerned with proving the feasibility of a symbolic model-based system for classification of militarily strategic terrain features from SAR imagery, and for long-term change detection of these features. The Phase I research concentrated on a rule-based methodology for extraction and classification of roads as distinct from competing features in SAR imagery.

This approach is unique in that it combines the relational, and qualitative attributes of known targets with the analytical / procedural approach of image-based processing. The symbolic model consists of a series of attributes that define a target feature. Determination of the model attributes activates image processing procedures automatically, as required, leading to an intricate classification result.

A prototype system was created that identified three road features from competing features. The prototype consisted of more than 120 rules which chained to over 20 image processing and analysis procedures. The operation of the system was based on a numerical scoring algorithm. For each cued feature, the knowledge base chained through various rules, requiring the execution of image processing sub-systems as necessary to determine the likelihood that the cued feature was one of the six features to be classified. Additional information was gained as required either by spawning procedures, or directly from the operator. As rules were executed a certaincy factor was updated according to the numerical scoring algorithm to indicate the likelihood that the candidate feature was a road.

The prototype was tested on 13 candidate features in four SAR data sets selected for the project. The test data sets comprised four SAR images from various sensors including SEASAT, JPL multi-polarization airborne SAR, and Intera Star-1 airborne SAR. The images provided a variety of terrain and targets, including ice floe images of the Beaufort Sea, agricultural regions in California, and urban scenes of Phoenix.

The results of this testing were successful over the four data sets indicating the feasibility of a semi-automated symbolic/procedure based system. The rule-based system correctly classified 71% of the candidate features. This is an excellent result for a fully automated classification system, since the initial rule system consisted on only 120 rules. This clearly shows the feasibility of such an approach. A highly detailed system containing many times more rules for many feature types should generate equally successful results. Such a system will be proposed for Phase II.

2.0 Introduction

The Phase I technical objectives are summarized in section 2.1, an overview of the conclusions appears in section 2.2, and the layout of the report is described in 2.3.

2.1 Background and Objectives of Phase I Research

The emphasis of the present effort was on the two main technical objectives for the development of a prototype system:

(1) Demonstration of descriptive adequacy of a rule-based approach to symbolic feature classification and change detection.

(2) Demonstration of adequacy of procedural image processing support routines for identification and segmentation of targets.

The Phase I objectives were achieved by creating a proof of concept multi-layer knowledge base with the capability of distinguishing six features commonly occurring in SAR imagery. The procedural adequacy was determined by manually executing image processing and polarimetry procedures as prompted by the knowledge system and feeding the results back into the inference engine.

At the direction of the TEC customer, the Phase I effort concentrated more strongly on development of the symbolic approach to create a working rule-based prototype, and providing partial development of key image processing subroutines and full specification of the less utilized image processing programs. That is, some of the image processing and image understanding procedures were fully specified including input, output and algorithms, but not coded. Also, the customer directed Vexcel to concentrate strongly on segmentation and identification of roads and road intersections during the Phase I effort.

The development of the prototype expert system lays the groundwork for a Phase II workstation that will incorporate rule-based classification, procedural registration and change detection, and high level image understanding algorithms together for effective target detection in hyper-spectral image data sets.

2.2 Summary of Conclusions

Results from testing the 120 rule prototype knowledge base indicate viability of the concept for automated target detection and identification in SAR imagery. The addition of multi-spectral, or hyper-spectral imagery would provide more information for a rule based system in Phase II, and a higher likelihood for success. Additional information provides more attributes for each feature class thus creating a more complete description for classification.

The prototype knowledge base was developed in Nexpert Object 1.1 on the Macintosh II for identification and classification of three types of roads from three competing features in SAR imagery. The prototype was tested on 13 candidate features from four SAR data sets including both satellite and airborne sensors. The JPL Raisin City data set included polarimetric data. The classification accuracy is summarized as follows:

Targets Classified Correctly:	71%
Targets Classified Incorrectly:	29%

Targets without classification frames in the prototype were classified with no bias in 50% of the cases, and incorrectly in 50% of the cases.

The primary reason for incorrect classification was ambiguity in the attribute description for each feature in the frame-based implementation of the inference engine. Also, when targets were provided to the inference engine for which no frames existed to describe them, they were incorrectly classified 50% of the time. Therefore it is important to develop frames for all possible features of interest that may be encountered in strategic situations. Further work is required to produce sufficiently distinct attribute sets for each feature of interest, possibly including hyper-spectral data attributes.

The knowledge environment should be more interactive and less automated. In the current system, potential exists for misclassification due to incomplete attribute descriptions in each frame. Even with highly detailed frame descriptions this problem will remain with fully automated systems. With a small amount of operator interaction, the incidence of incorrect classification could be radically reduced without much time lost.

More descriptive frames for feature classes of interest should be developed. Frames for unwanted classes can be sparser as long as they are sufficient to remove such features from consideration as features of interest.

The Phase II workstation should consist of an interactive environment in which the user may communicate with the inference during processing. This will be an effective combination that will be more efficient than an operator alone, and produce more accurate results than the expert system alone.

There is a tradeoff between development of a broader base for classification of a wide variety of features vs. development of highly detailed descriptions of fewer strategic targets. In the former, each class would be described by relatively sparse frame attribute sets. Misclassification would be more prominent, and the resulting system would essentially provide a demonstration of the possible applicability of symbolic systems in classification and change detection. It is therefore recommended that the Phase II effort concentrate on extensive symbolic and procedural development of one class of features. The system architecture will be designed to accommodate numerous feature classification knowledge systems, but the bulk of the time should be directed toward producing a useful tool for automated classification and change detection.

2.3 Organization of Report

The remainder of the report is organized as follows. Section 3 describes the organization of the knowledge, operation and control of the overall system. Development methodology of the rule base is detailed including the breakdown of knowledge using both decision trees and analysis of competing features. Section 3 also discusses the operation of the control executive, issues associated with rule based systems and certainty factors, and the usage of a non-binary scoring function to keep track of feature likelihoods. Section 4 describes the image processing procedural interface and requirements. All required image processing routines used in the prototype are described in section 4 including I/O specifications and algorithms. Section 5 describes the rules. Both the "meta-rules" and the encoded Nexpert Object 1.1 rules are discussed here. Reasoning and factual basis of each rule is discussed and tradeoffs between exact rules system and exact image processing systems are detailed. Section 6 describes results from tests on various SAR images from differing SAR scenarios. Section 7 is comprised of a complete explanation of results and conclusions, and Section 8 provides recommendations for Phase II.

3.0 Organization of Knowledge

In a rule-based system, the knowledge representation scheme dominates the formulation of the system. The organization of the rules can be more important than the rules themselves. The infrastructure of the system must impose strict limitations on how new information is added and how it may affect other data in the knowledge base. A more in depth discussion of symbolic systems follows in Section 3.1

The first manifestation of such a strict organization of knowledge, is that the information must be broken down into specific domains. Secondly, the rules should be organized into groups such that each group is concerned with one or a class of similar features. This will ensure that similar competing features in the target images will be treated by the same rules so that small differences between competing objects can be used to advantage in the inference process. Section 3.2 further discusses the breakdown of knowledge by these two approaches.

Unlike procedural systems, in rule based systems, the boundary between logic and control is clearly defined. The inference engine seeks to control the flow of computations by chaining from one rule to another, while the knowledge is contained in the rules themselves. When dynamically linked with a procedural system, an executive controller is required to dictate the order in which sub-areas of knowledge are accessed. As a result, the inference engine should be managed by a *control executive*. The executive will start, communicate with, and terminate the inference engine. Also, the control executive will handle the interface between the inference engine and the procedural sub-systems. The requirements of the control executive are addressed in Section 3.3.

During the inference process, the effect of each rule must be traced. If the conclusion is not clearly true or false, a non-binary value or score may be used to track the certainty of the classification of the feature being processed. The issues and requirements for a scoring function are addressed in section 3.4.

3.1 Discussion of Symbolic Systems

A symbolic classification system or model is one that uses relational, qualitative, and quantitative aspects to describe an object. Symbolic systems are more general than procedural systems which use only derived quantitative measures for classifying an object. A procedural system can be a component of a symbolic system. Symbolic systems may take a multitude of forms and there are a number of ways to approach the SAR feature classification and change detection problem.

In the development of any AI system, the important issues concern representation, implementation, control and uncertainty. Section 3.3.1 discusses two alternative representations of domain-specific, "expert" knowledge. Section 3.1.2 contains a brief discussion of the serial invocation of such expert modules by a procedure. A multiple-pass implementation is considered in section 3.1.3. Finally, the use of a numerical measure of certainty of conclusions is introduced in 3.1.4. This subject is discussed more fully in 3.4.1.

3.1.1 Rules vs. Frames

Rule based processing, sometimes called a *production system*, represents a formal method using the predicate calculus to infer new facts from those known already. It is in essence automated theorem proving applied to more heuristically understood application domains. In this way, knowledge about a certain domain is characterized by all of the assertions which can be proven from an initial *knowledge base* which is encoded as a set of rules.

The "if (premise) - then (conclusion)" paradigm of rule-based processing has certain advantages. One can add, modify, or delete rules easily, so from a programming standpoint it is a convenient structure. On the other hand, one cannot easily access the knowledge represented by a set of rules all at once. Moreover, combinations of rules will generally come up with unanticipated conclusions. This is sometimes advantageous and often disadvantageous.

A brief overview of some of the deeper problems associated with the use of rules appears in section 3.4.1.2.

The notion of a *frame* was introduced in [Minsky, 75]. It represents a more heuristic approach to knowledge representation. Frames can be thought of as models for "prototypical" situations. Essentially, frames attempt to collect all of the information on a given subject, object, or situation, into a single structure in terms of its properties. As such, it has similarities to the notion of *object-oriented* programming.

In this way, a frame is a network of nodes and relations, where the top levels represent generalities of *default assignments*, i.e. the things which are always true about a situation, and the lower levels are *slots* which are filled by more specific instances. A major issue is the extent to which subclass specificities should overrule class-defaults in any given situation.

To illustrate the difference between rules and frames, we present a few examples. Rules are concerned with direct relationships that equate one situation to another. For example, in a diagnostic system:

> *If temperature is greater than 4000° then reactor has overheated.*
> *If reactor has overheated then pressure release valve should be opened.*

These rules equate the situation "temperature > 4000°" to "reactor has overheated", at which point the term "reactor has overheated" can be used later. Also, there is a link defined between "reactor has overheated" and temperature. If there is a question about opening the pressure release valve, a rule system could backward chain through ""reactor has overheated" to check the temperature of the reactor. Rules alone can be very powerful in a situation in which there are many direct relationships.

The important thing to notice with the example above is that "reactor has overheated" is equivalent to "temperature > 4000°". There is no question that if the temperature exceeds 4000° whether overheated is true or not. In cases where there is not a direct relationship, rules require more information. For example, the fact, "Trees have leaves," may be converted to a rule as:

If an object is a tree then it has leaves.

As above, this rule equivalences "object is a tree" with "it has leaves". This, however, is not always true. Is a cactus, for example, classified as a tree? Are pine needles classified as leaves? Is it winter? Clearly, leaves are an attribute of some, but not all trees. Depending on whether the object of the rule is to classify leaves or trees, two different "frame-like" approaches can be taken. Assuming leaves are to be classified, the following attribute oriented rules may be used.

If an object is a tree then it has leaves with certainty factor 85%.
If an object is a bush then it has leaves with certainty factor 60%.
etc.

Of course, these rules assume tree and bush are already defined, or that "object is a tree" is a conclusion of some other rule so that backward chaining may occur.

If tree is the object to be classified, then, as above, exceptions to the fact "Trees have leaves" must be investigated. For example, bushes have leaves, flowers have leaves, etc. Thus, an appropriate "frame-like" rule for classification of trees would be:

If an object has leaves then it is a tree with certainty factor 60%.
If an object has bark then it is a tree with certainty factor 95%.

This "frame-like" approach consists of a list of attributes, possibly as rules, that drive a certainty factor toward one extreme or another. One advantage of this method is that frame attributes need not be all inclusive. Only enough attributes need be included to define an object well enough that it can be differentiated from other object frames. Thus if a system consisted of only two frames, as few as one attribute per frame would be required, if those attributes were mutually exclusive. For example, if the system were to classify trees and bushes, the attribute "has leaves" would not be sufficient to differentiate the two classes.

The present implementation will be seen to combine aspects of both rules and frames. Here, rules are used to implement the geometric and radiometric descriptions of the signatures of each type of terrain feature which is either desired, or similar to those features of interest. In this way, each major terrain feature has its own frame. Each type of potential feature interpretation will have a numerical score which was computed using the evidence presented. The preferred interpretation will then be for the feature with the highest score.

3.1.2 Serial Invocation of Expert Systems

Another methodology for approaching this classification problem concerns a system in which expert system modules are called in series for each feature type to evaluate a portion of the image data which had been previously cued by procedural *interest operators*. In this case, there would have to be a separated expert system module for each different feature type. Again, a numerical score would be used to find the probable interpretation of the evidence extracted from the imagery.

Such an approach may have advantages in modularity, but is not as easily able to accommodate knowledge which cuts across possible interpretations. One could conceive of rules whose premises include the uncertainties associated with conclusions of alternative interpretations.

For example, one such rule might be:

If (score(interpretation A)+score(interpretation B)) ≤ (score_threshold) then update the score for interpretation C by certainty factor p_C.

3.1.3 Multistage Expert System Evaluation

Another heuristic which has not been employed in the present system is the potential dependence of a given interpretation in one image region on that obtained in another region. For example, roads are part of connected networks which are not isolated from each other. Generally, one can travel from one road to any other in a region.

Therefore, one could imagine a situation in which the interpretation for a feature f_1 as being a "road" is ambiguous in region R_1, but very strong for feature f_2 in region R_2. If a connected path of "road" can be subsequently found to connect f_1 with f_2, then the strength of interpretation of "road" for f_1 should strongly increase.

Because of time limitations, such a strategy could not be considered for the Phase I effort.

3.1.4 Numerical Evaluation for Possible Interpretations

Clearly, all conclusions based on measurements have inherent uncertainties. In this system, most individual measurements will only contribute a marginal amount of evidence toward a particular interpretation for a previously cued, potential feature.

It is not clear how human aggregate partial evidence and come to a conclusion. Certainly introspection does not reveal any numerical basis for such processing, even if there is one. However, most evaluation methods used in AI use some type of numerical computation.

One notable exception is a non-numeric logic for manipulating sentences containing qualifiers that is described in [Halpern et al, 87] and [Fagin et al, 88]. However, there are still considerable remaining technical problems for this approach.

Therefore, we have preferred a method for incrementally accumulating numerical scores of multiple interpretations for image features as more images-based evidence is obtained. This method is a modification of the heuristic *certainty factors* used in MYCIN, [Shortliffe, 76], and is discussed in section 3.4.1 along with other methods.

3.2 Orthogonal Knowledge Framework

The rule development effort used a two-pronged approach. Initially, the effort was concerned with the development of a breakdown of the areas of knowledge to be addressed in the rule base. Secondly, we approached the identification of features of interest with respect to similar features likely to be visible in the SAR that would compete with the desired features. These two methods are independent means of organizing the knowledge. As a result, the two can be used to create an orthogonal structure that divides up the knowledge of the system into very exact niches. The next two sections describe the two orthogonal methods.

3.2.1 Knowledge Breakdown by Attributes

The initial breakdown by attributes proceeded by identifying rule divisions for SAR scenarios. The assumption here is that in the future, the knowledge base may incorporate types of images other than SAR for automated target recognition. Under each sensor scenario, the areas of knowledge are successively broken down until only the properties of each sub-area are listed. For example, under the imaging physics of the SAR scenario, the properties include: wavelength, resolution, signal to noise ratio, look angle, squint angle, etc. This list of properties constitutes the *leaves* of a the organization tree. It is at the leaves, only, that knowledge oriented rules are placed.

Rules can then be added to each leaf using only the properties of that area. The effects of the instantiation of each rule are also limited to the appropriate leaf or nearby leaves with intersecting properties. If there are likely to be many rules that incorporate more than one area of knowledge, new leaves are created in the organization for them. Through this methodology, fewer rules are repeated, and fewer conflicts occur as the size of the rule base increases. This breakdown for the prototype is shown in Figures 3.1 to 3.4 Rules may exist at higher nodes in the tree, but such rules would affect control of the chaining mechanism between knowledge rules, rather than the actual knowledge itself.

After the initial development of the top-down rule organization just described, several meta-rules were developed that used the physics of the imaging scenario to extract further knowledge from the imagery and current information. A meta-rule, is an assemblage of bits of knowledge into an *if-then-else* format. Generally, it is not possible to directly use or encode meta-rules since it may include non-exact terminology such as *"if the feature is characterized by a partially curvilinear signature then ..."*. Clearly, terms such as *partially* and *curvilinear* must be defined in a quantitative manner, and the rule must be refined such that it succinctly determines the requirements. But, the use of meta-rules provides an initial means for qualitatively defining measures of information. Meta-rules can generally be coded into an expert system using one to six actual rules. A complete explanation of all rules and their reasoning basis is given in section 5, but for illustration an example of an initial meta-rule and its reasoning is given below:

If *(indicated region is characterized by a bimodal histogram) and*
(the statistically darker region is characterized by a closed contour
or

a contour that intersects the down-range edge of the region)
Then (shadowed region is confirmed) and
(vectorized border of the region can be defined).

The reasoning behind this rule is that a shadowed region in a SAR image is characterized locally by a bimodal histogram. The identification of shadowed regions is important in the identification of roads because terrain highlighting effects associated with ridges or mountains are often adjacent to shadows. Because of their bright signature and linear characteristic, terrain highlighting effects may be confused with road features by the knowledge system. As a result, if a candidate feature is adjacent to a shadow along a large percentage of its length, that feature can reasonably be classified as a terrain effect.

3.2.2 Knowledge Breakdown by Feature Type

Another method for organizing rules that recognize a small group of features in a SAR image is to identify all types of signatures visible in the image that might be similar in some way to the desired features. In particular, certain image processing functions are likely to identify some competing features along with features of interest. This organizational method concentrated on determining which characteristics of features impact the segmentation of various image processing procedures. For example, global intensity thresholding is likely to segment terrain highlighting effects as well as divided highways. But, terrain highlighting effects are also likely to be adjacent to shadows, and divided highways may be characterized by local regional contrast.

This method seeks to identify the fact that divided highways and terrain highlighting effects are similar competing features. In addition, it is noted that adjacency to shadow, and local regional contrast are image processing routines that can differentiate between the features, while global intensity thresholding, for example, cannot.

Along this line of reasoning more than 20 target features were identified, and 28 distinguishing characteristics were identified. The features were identified both from the test images described in section 6.1, and from a catalog of digital feature analysis data (DFAD). The geometric signatures, textural and other characteristics were cataloged. Subsequently, distinguishing characteristics were codified from the signature descriptions and relationships between these characteristics, the target features, and the required image processing sub-systems were identified. Each distinguishing characteristic was identified to be associated with each feature on an always, usually, sometimes, rarely or never, basis. See Table 3.1. This methodology provided a non-binary measure of the usefulness of each characteristic with respect to identification of each feature. Then when each characteristic was associated with an image processing function. The procedural definition of the system was completed for all named features. That is, the image processing procedures were defined that could be used to distinguish each of the 20 selected features. Table 3.2 shows the relationship between the image processing routines, and the distinguishing characteristics.

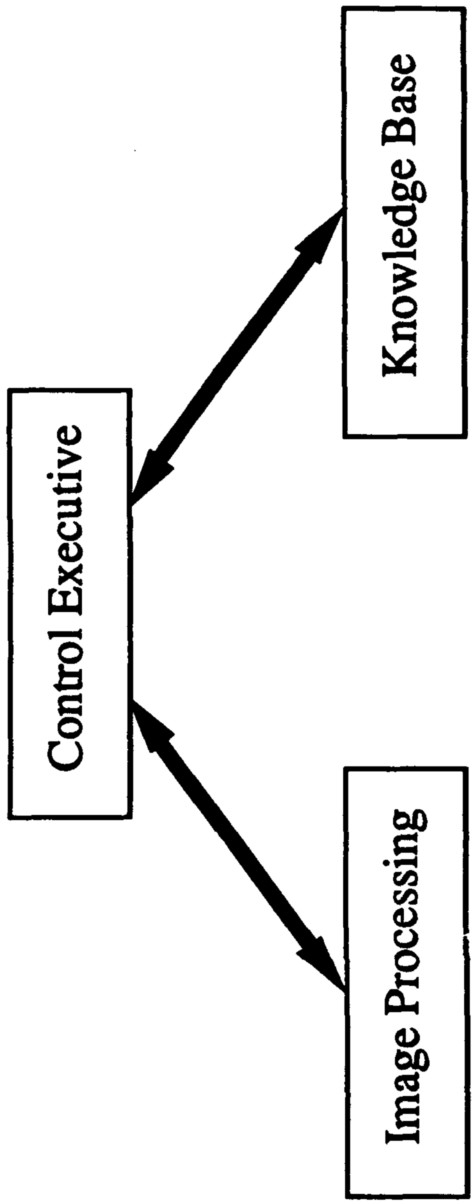


Figure 3.1 Initial breakdown between knowledge and control.

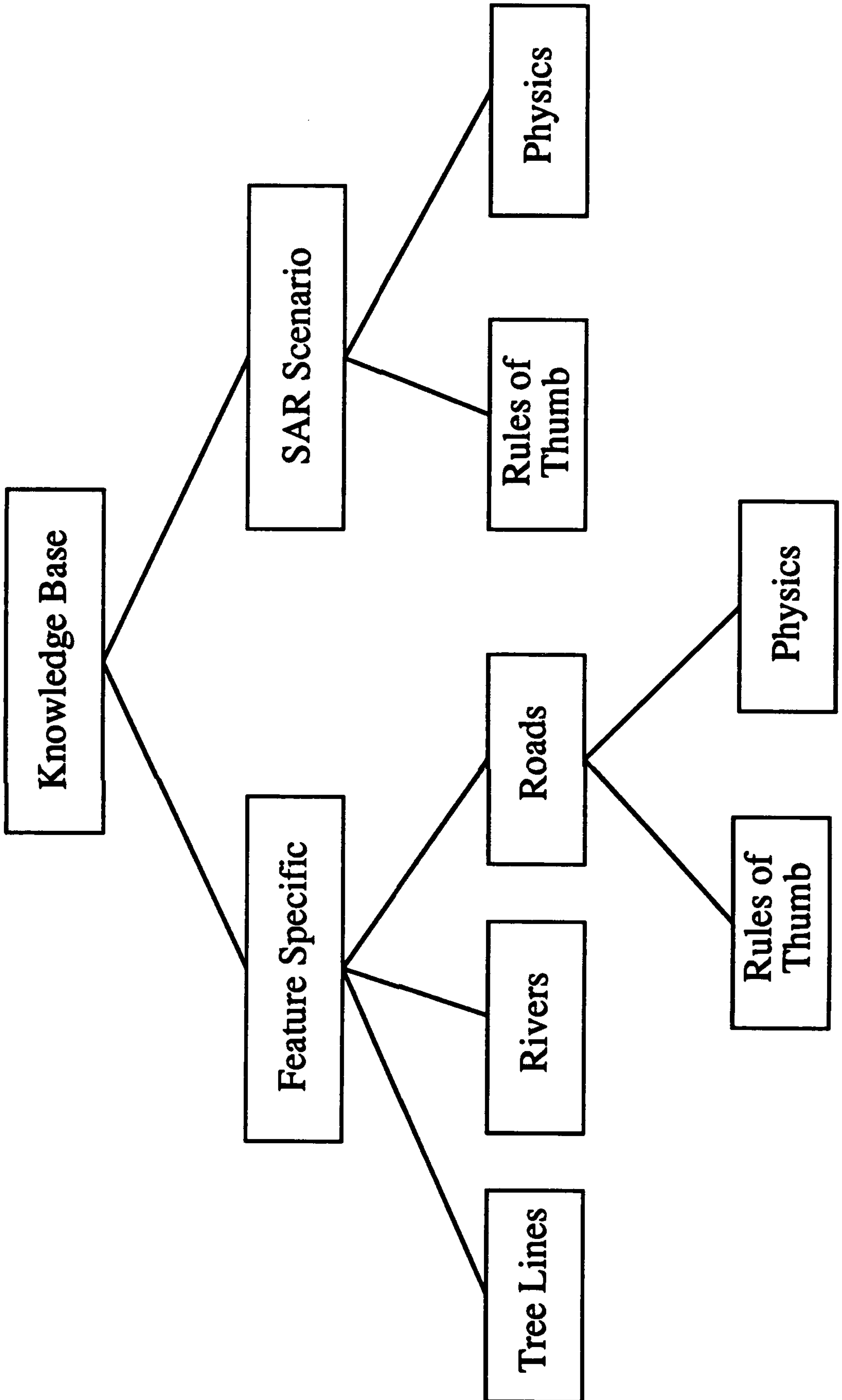


Figure 3.2 Organization by areas of knowledge of the rule base.

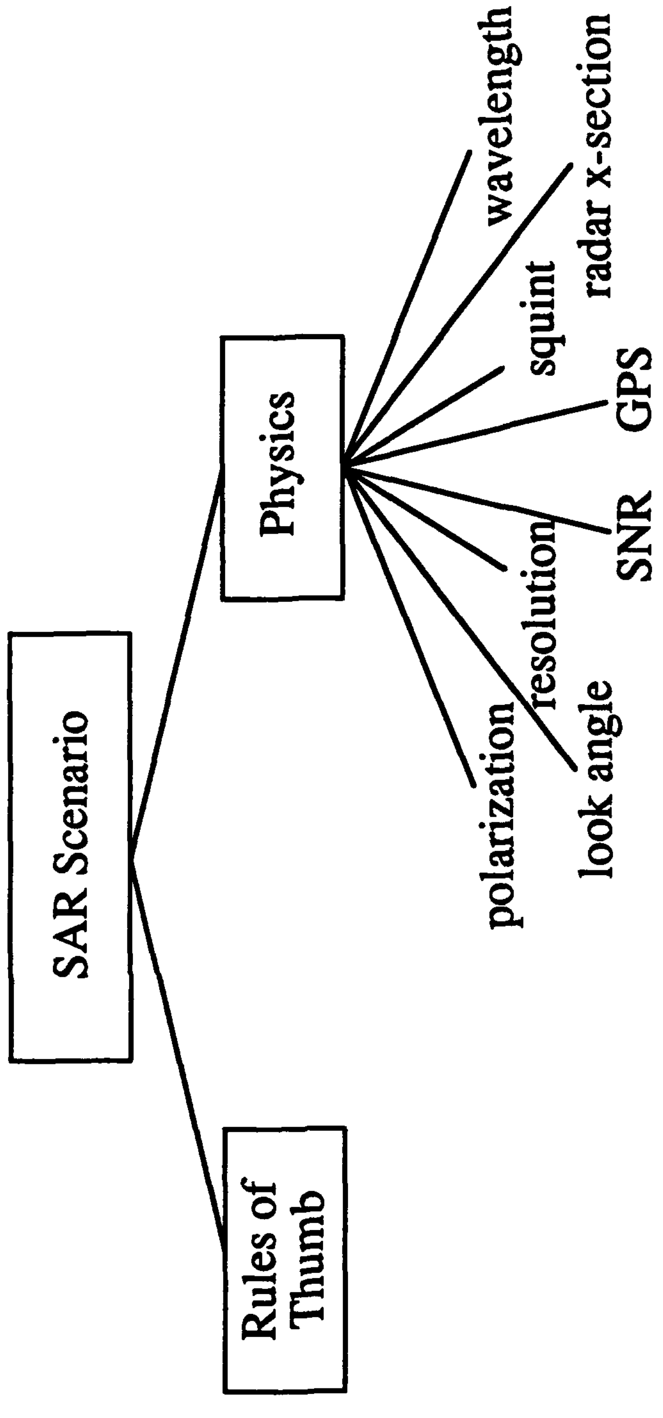


Figure 3.3 Organization of SAR Scenarios by areas of knowledge.

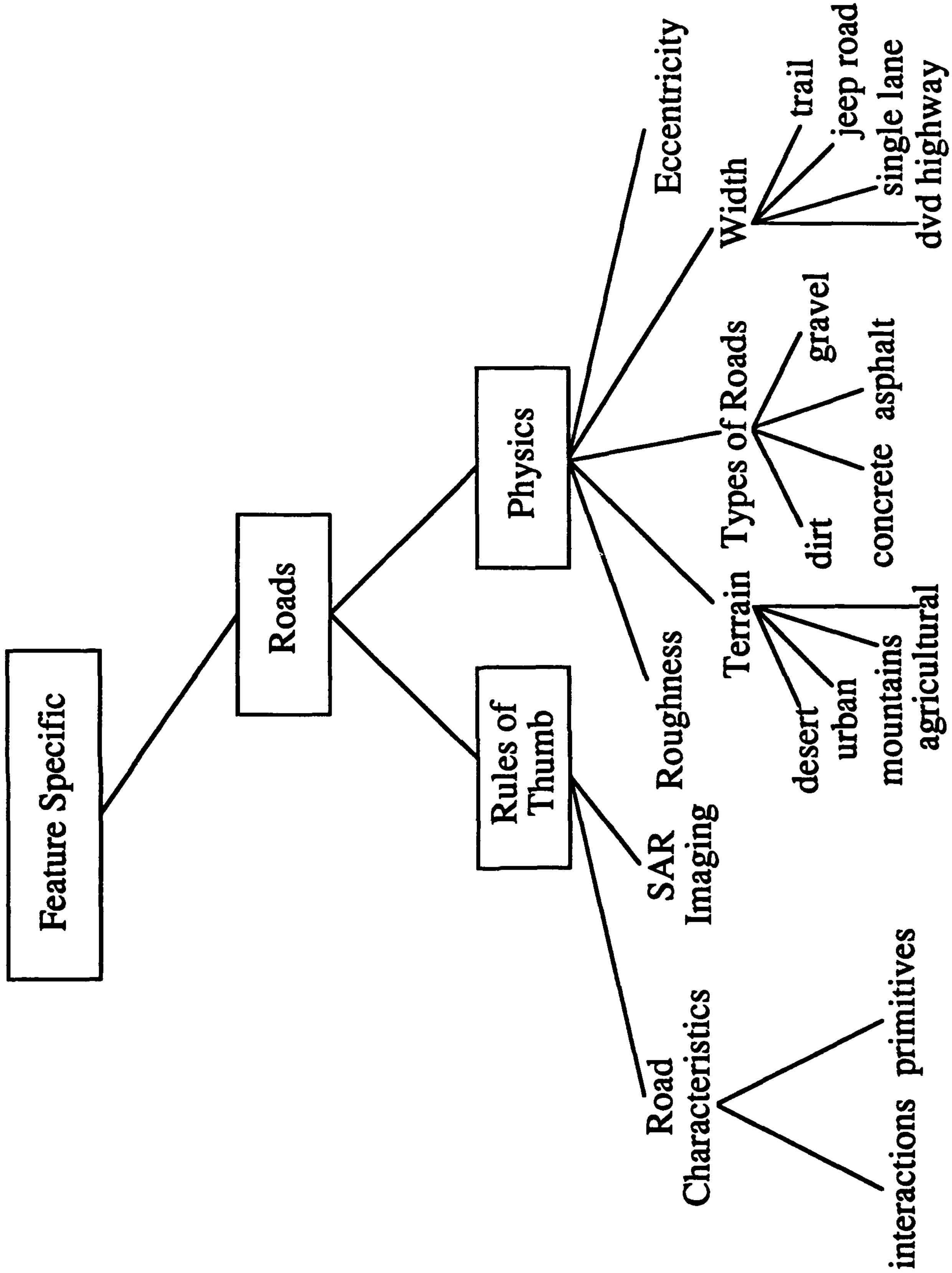


Figure 3.4 **Organizational breakdown of feature specific knowledge.**

Table 3.1 Relationship between distinguishing characteristics and feature classes.

Legend
- X = always
- + = usually
- O = sometimes
- ● = rarely
- = never
- — = inconclusive
- L = low
- H = high

	linear	curvilinear	gravel	asphalt	concrete	dirt	stone	brick	water content	# lines	small std. dev. (avg spacing)/resolution	# of lanes of each line	orientation with compass direcitions	motion	local contrast	edge gradient	intensity of backscatter	small std. dev. of intensity along feature	bimodal histogram	Rayleigh distributed histogram	odd # bounces	even # bounces	high density of corner reflectors	dark lines in clutter	dark regions in clutter	adjacent to shadow	interconnections
cloverleaf interchange	X	X		+	+				L	2+	X	2+	O		+	+	—	+	—	O	—	—	O	O	+		X
limited access interchange	X	O		+	+				L	2+	X	2+	O		+	+	—	+	—	O	—	—	O	O	+		X
trumpet interchange	X	X		O	O				L	2+	X	2+	O		+	+	—	+	—	O	—	—	O	O	+		X
urban grid	X			O	O	O	O	O	—	4+	X	2+	X		+	O	H	+	O	O		X	X	X	+		X
cross intersection	X		O	O	O	O	O	O	—	1+	X	2+	O		+	+	—	+	—	O	—	—	+	+			X
T intersection	X		O	O	O	O	O	O	—	1+	—	2+	O		+	+	—	+	—	O	—	—	+	+			X
traffic circle		X	O	O	O	O	O	O	—	1	—	1+	—		+	+	—	+	—	O	—	—	+	+	+		X
forest road	O	O	O			O			—	1	—	1			O	O	—	+	—	—	—	—	—	—	—		O
secondary road	O	X		O	O				L	1	—	2	O		+	+	—	X	—	—	—	—	O	O			+
divided highway	O	X		O	O				L	2+	X	2+	O		+	+	—	X	—	—	—	—	O	O			+
div. hwy. w/serv. rd.	O	O		O	O				L	3+	X	2+	O		+	+	—	X	—	—	—	—	O	O			+
causeway	O	O		O	O				—	1	—	2+	●		X	X	H	X	+	O		X				+	●
road and river	O	+	O	O	O	O	O	O	—	2+	—	2	●	X	+	+	—	+	—	—	—	—	●	●			
dam and road	+	O	O	O	O	O	O	O	—	1	—	2	O	O	O	X	—	O	+	O	—	—	●	●		O	
lake and road	O	O	O	O	O	O	O	O	—	1	—	2	O	O	O	O	—	O	+	O	—	—	●	●			
crops and road	X	O	O	O	O	O	O	O	—	1+	—	2	O		O	O	—	O		+	X						
airport	X			O	O				L	1+	+	4+	+		+	+	L	X	O	O	O	O	X	X	X		O
tree boundary	O	O							—	1	O	—	O		O	X	—	●		O	—	—					
crop boundary	X	O							—	1+	O	—	O		O	X	—	●		O	X	—					
river	O	O							H	2+	●	—	●	X	+	X	L	●	O	O	+	—	X	X	+		X
river confluence	O	O							H	4+	●	—	●	X	+	X	L	●	O	O	+	—	X	X	+		X
clear cut	X								—	1	O	—	O		+	+	—	O		O	●	+	+	+			
transmission line	X								—	1	O	—	O		+	+	—	O		O	—	—	+	+			
canals		O							H	2	O	1	O	X	+	+	—	O		O	—	—	+	+			●
aquaducts	X								H	2	O	1	O	X	+	+	—	O		O	—	+	+	+			●
swampy region	O	X							H	—		—		X	●	●	—		+	●	—	X			+		X
shoreline		X							H	1		—		X	●	X	L		+		—	+			+		
terrain higlighting effects	O	O	—	—	—	—	—	—	—	1	X	—	O		X	X	H	+	O	O	—	—				X	
SAR shadow	O	O	—	—	—	—	—	—	—	1		—	—			+	L	—	X		—	—	X		X	—	—
antenna side-lobe effects	X		—	—	—	—	—	—	—	3+	●	—	O		●	O	—	—		X	—	—				—	—
speckle	—	—	—	—	—	—	—	—	—	—		—	—			O	—			X	—	—				—	—

Table 3.2.2 Relationships between image processing primitives and feature distinguishing characteristics.

Local Tangent	Intensity Variation	Mathematical morphology	DEM surface analysis	Streak detectors	Edge operators	Local lineal statistics	Local Regional statistics	Analysis of Meuller matrix	Autocorrelation	Region Grower	Interconnection detection	
									×		×	linear
									×		×	curvilinear
							×			×		gravel
							×			×		asphalt
							×			×		concrete
							×			×		dirt
							×			×		stone
							×			×		brick
			×			×						water content
							×				×	# lines
					×		×					small std. dev. (avg spacing)/resolution
							×		×			length
					×						×	# of lanes of each line
									×			orientation with compass direcitions
								×				motion
							×					local contrast
						×						edge gradient
										×		intensity of backscatter
					×							small std. dev. of intensity along feature
				×								bimodal histogram
				×								Rayleigh distributed histogram
			×									odd # bounces
			×									even # bounces
			×	×								high density of corner reflectors
		×				×	×					dark lines in clutter
	×			×								dark regions in clutter
				×				×				adjacent to shadow
×												interconnections

Next, a rule structure was developed that dynamically initiated image processing as necessary to affirm or deny that a cued object was a feature of interest (road). To accomplish this, each image processing function was ranked according to its impact on the probability of determining the classification of a cued feature. From this ranking, rules were formed that impacted the likelihood of classification. For example:

> *If (feature avg. intensity is greater than a given threshold) and*
> *(cued feature consists of a single linear signature) and*
> *(feature is adjacent to a shadowed region)*
> *Then (likelihood cued feature is a terrain effect) should be increased*
> *Else (likelihood cued feature is a terrain effect) should be decreased.*

These rules were ordered from most important to least important for each feature. For the prototype system, six features were selected from the larger group. The selected features were ones that were likely to compete, and to provide a good test for the formulation of the knowledge system. The six selected features included three road features and three competitors, as follows: urban road grid; divided highway; unimproved road; terrain highlighting effect; agricultural field boundary; and transmission line. Using the preferentially ordered image processing routines and measures of the characteristics associated with each, a decision tree structure was created to test each cued feature under all possible circumstances. Various paths through each decision structure resulted in differing likelihoods of classification for each selected feature type.

The decision tree structure for the divided highways is shown in Figure 3.5, other decision trees are shown in the Appendix. Paths through the decision trees are determined by the Boolean value of the result of each test. TRUE results result in downward sequence, FALSE in sequence to the right. Note that the same tests are executed regardless of TRUE or FALSE evaluation , but that the rule strengths, and therefore the updated certainty factor at each point are different. Figure 3.5 provides a useful tool in the development of a rule structure in that it indicates the required image processing sub-programs required for each test, and the rule strengths to be assigned to those tests to update the certainty factor.

The decision trees were used to create a prototype rule network shown in Figure 3.6. The prototype system consists of six layers corresponding to each selected feature class. Depending on the value of the rule strength for each rule after it is instantiated, one of two paths may be taken. The processing may continue evaluating the current feature, or it may chain into the evaluation sequence of another feature. The ordering of the layers in Figure 3.6 is arbitrary, and in the actual prototype implementation, the inference engine is free to chain to any connected rule at any time. This means that it can evaluate the likelihoods for several features at the same time chaining from rules concerning divided highways, for example, to ones concerning transmission lines and back again. A more concise description of the reasoning behind each rule and the architecture of the prototype rule system are described in Section 5.

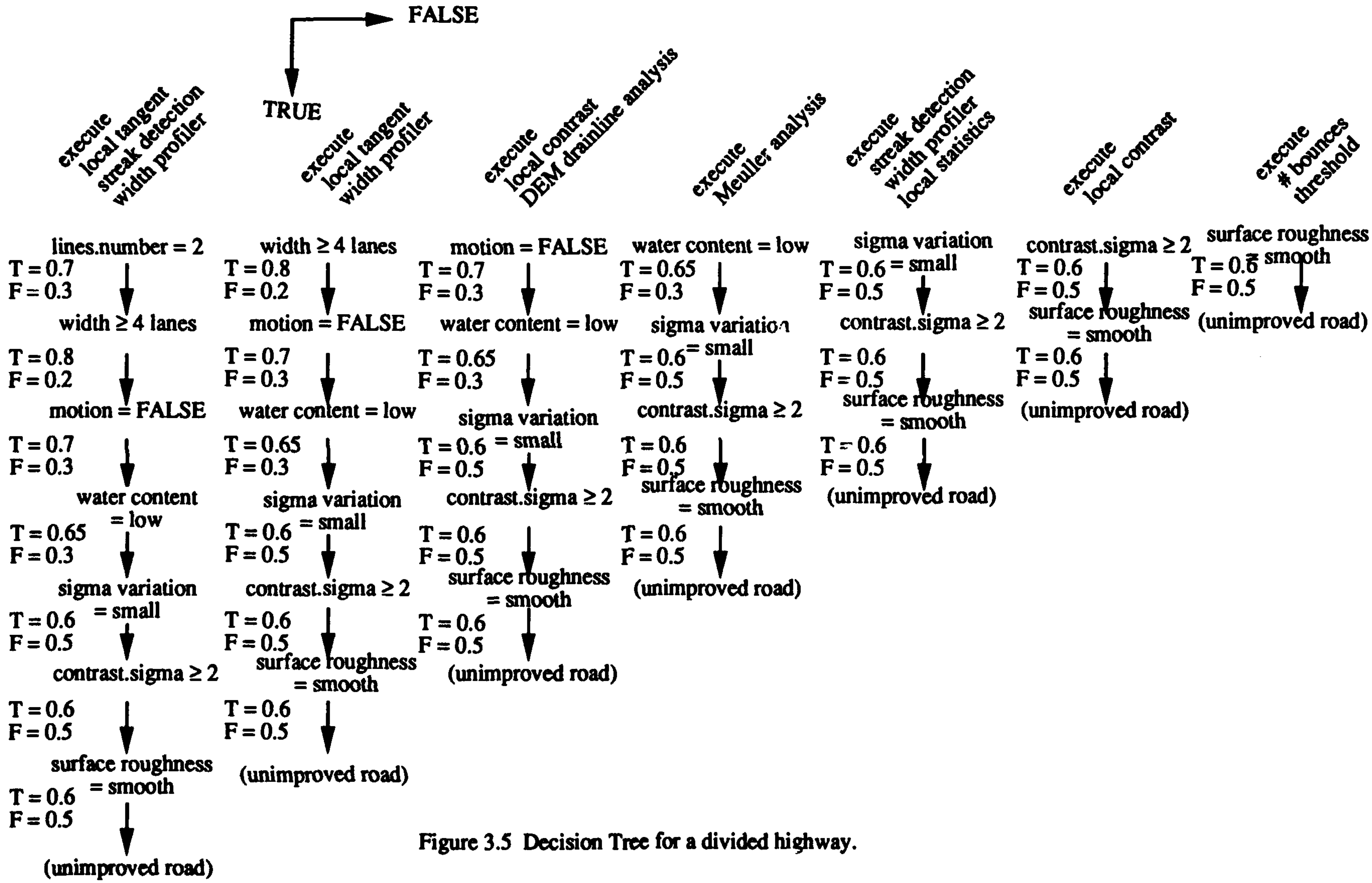


Figure 3.5 Decision Tree for a divided highway.

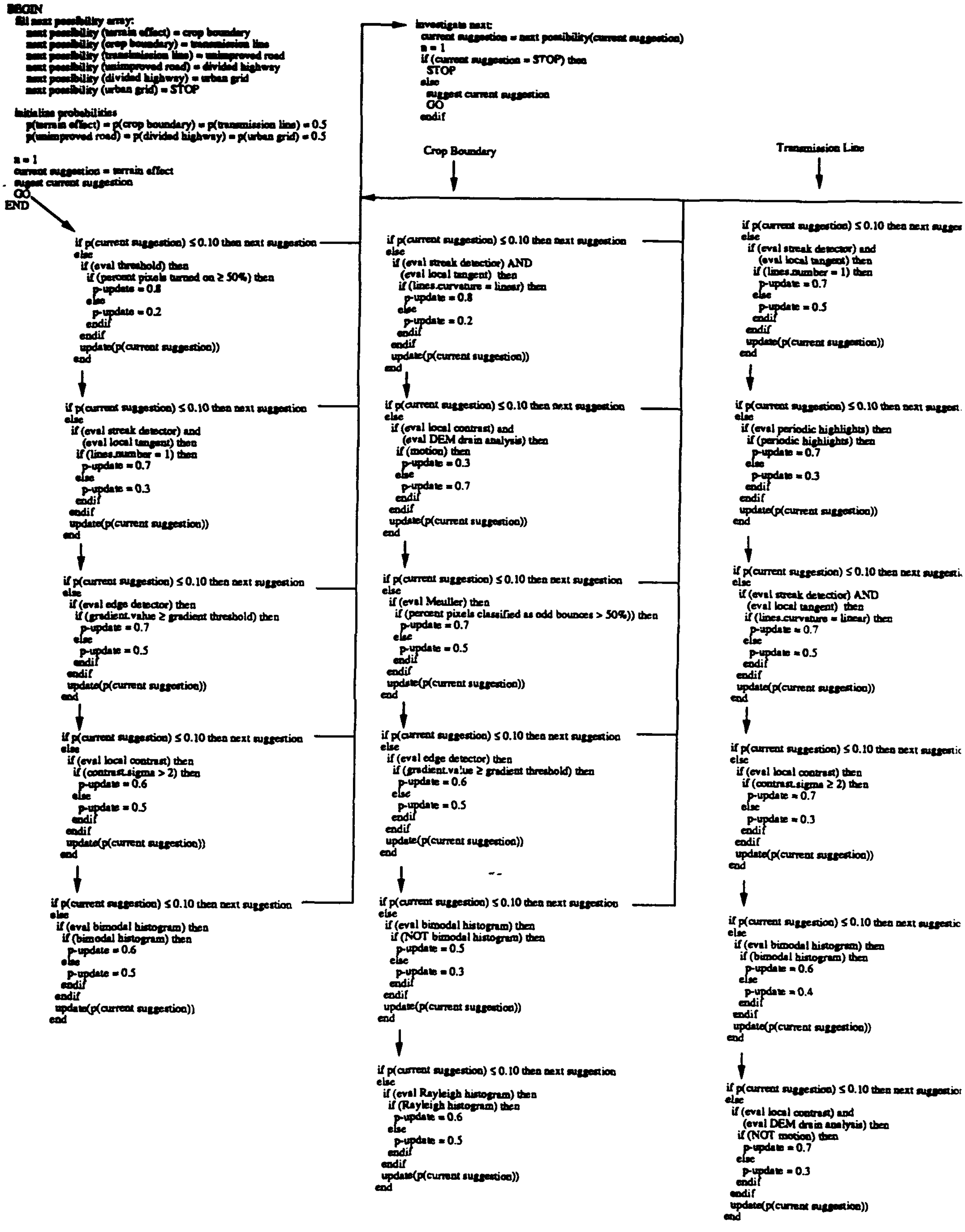


Figure 3.6 Rule network derived from decision trees. The image processing modules named here may be cross-referenced in Chapter 4.

```
) next:
uggestion = next possibility(current suggestion)
t suggestion = STOP) then

current suggestion
•
```

```
undary

gestion) ≤ 0.10 then next suggestion

detector) AND
ngent) then
ture = linear) then
.8

.2

nt suggestion))

estion) ≤ 0.10 then next suggestion

ntrast) and
rain analysis) then
n
.3

.7

t suggestion))

estion) ≤ 0.10 then next suggestion

) then
ls classified as odd bounces > 50%)) then
7

5

t suggestion))

estion) ≤ 0.10 then next suggestion

ector) then
ue ≥ gradient threshold) then
6

5

t suggestion))

stion) ≤ 0.10 then next suggestion

histogram) then
al histogram) then
5

t suggestion))

stion) ≤ 0.10 then next suggestion

histogram) then
togram) then

suggestion))
```

Transmission Line

```
if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 1) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval periodic highlights) then
    if (periodic highlights) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) AND
     (eval local tangent) then
    if (lines.curvature = linear) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local contrast) then
    if (contrast.sigma ≥ 2) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval bimodal histogram) then
    if (bimodal histogram) then
      p-update = 0.6
    else
      p-update = 0.4
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local contrast) and
     (eval DEM drain analysis) then
    if (NOT motion) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end
```

Unimproved Road

```
if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 1) then
      p-update = 0.6
    else
      p-update = 0.0
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval Meuller analysis) and
     (eval threshold) then
    if (surface.roughness = rough) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local tangent) and
     (eval width profiler) then
    if (width ≤ 2*lane.size) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval threshold) and
     (eval Meuller analysis) then
    if (density.corner_reflectors = low) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval bimodal histogram) then
    if (bimodal histogram) then
      p-update = 0.5
    else
      p-update = 0.6
    endif
  endif
  update(p(current suggestion))
end
```

Divided Highway

```
if p(current suggestion) ≤ 0.10 then n
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 2) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then n
else
  if (eval local tangent) and
     (eval width profiler) then
    if (width ≥ 4*lane_size) then
      p-update = 0.8
    else
      p-update = 0.2
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then ne
else
  if (eval local contrast) and
     (eval DEM drain analysis) then
    if (NOT motion) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then ne
else
  if (eval Meuller analysis) then
    if (water content = low) then
      p-update = 0.65
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then ne
else
  if (eval streak detection) and
     (eval width profiler) and
     (eval local lineal statistics) then
    if (intensity.sigma ≤ low_sigma_thr
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then ne
else
  if (eval local contrast) then
    if (contrast.sigma ≥ 2) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then ne
else
  if (eval Meuller analysis) and
     (eval threshold) then
    if (surface.roughness = smooth) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end
```

ision trees. The image
s-referenced in Chapter 4.

2

Unimproved Road

```
if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 1) then
      p-update = 0.6
    else
      p-update = 0.0
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval Meuller analysis) and
     (eval threshold) then
    if (surface.roughness = rough) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local tangent) and
     (eval width profiler) then
    if (width ≤ 2*lane.size) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval threshold) and
     (eval Meuller analysis) then
    if (density.corner_reflectors = low) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval bimodal histogram) then
    if (bimodal histogram) then
      p-update = 0.5
    else
      p-update = 0.6
    endif
  endif
  update(p(current suggestion))
end
```

Divided Highway

```
if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 2) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local tangent) and
     (eval width profiler) then
    if (width ≥ 4*lane_size) then
      p-update = 0.8
    else
      p-update = 0.2
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local contrast) and
     (eval DEM drain analysis) then
    if (NOT motion) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval Meuller analysis) then
    if (water content = low) then
      p-update = 0.65
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detection) and
     (eval width profiler) and
     (eval local lineal statistics) then
    if (intensity.sigma ≤ low_sigma_threshold) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local contrast) then
    if (contrast.sigma ≥ 2) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval Meuller analysis) and
     (eval threshold) then
    if (surface.roughness = smooth) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end
```

Urban Grid

```
if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detector) and
     (eval local tangent) then
    if (lines.number = 2) then
      p-update = 0.85
    else
      p-update = 0.15
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local tangent) and
     (eval width profiler) then
    if (width ≥ 4*lane_size) then
      p-update = 0.65
    else
      p-update = 0.35
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval streak detection) AND
     (eval local tangent) then
    if (lines.curvature = linear) then
      p-update = 0.7
    else
      p-update = 0.3
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local contrast) then
    if (contrast.sigma ≥ 2) then
      p-update = 0.6
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval local tangent) and
     (eval width profiler) and
     (eval streak detector) then
    if (lines.spacing_variance ≤ line spacing threshold) then
      p-update = 0.9
    else
      p-update = 0.1
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval Meuller) then
    if (percent pixels classified as even bounces > 50%) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end

if p(current suggestion) ≤ 0.10 then next suggestion
else
  if (eval region grower) then
    if (execute region anlaysis yields dark regions in clutter) then
      p-update = 0.7
    else
      p-update = 0.5
    endif
  endif
  update(p(current suggestion))
end
```

3.3 Control Executive

The Control Executive is the supervisor of the rule/procedure system. The control module handles all interaction between the reasoning system and the image processing modules. In addition, it is the control executive that is responsible for system level interactions of the knowledge base. That is, it must handle memory allocation, memory I/O, and disk I/O for the rule system. When the rule system requires information from image processing routines, or other files on disk, it sends a message to the control executive which then, procedurally, carries out the request and returns the required information to the rule system. This requires that the control executive be able to extract information from and place it into the registers of the rule base.

The requests from the rule base can be encoded simply, by using a lookup table of codes. For example, when the Sobel edge detector must be called, the rule base may pass a simple numerical code to the control executive module. At that point the control executive may use the code to look up the appropriate procedure. The procedure will then extract any necessary data from the knowledge system, call the image processing subroutine, write the required information back into the knowledge base, and signal the knowledge system of successful completion.

Section 3.3.1 describes the Nexpert Control System that was used during the Phase I effort, and Section 3.3.2 describes the proposed control executive to be used in the Phase II effort.

3.3.1 Nexpert Control Executive

The Nexpert Object 1.1 development system consists of a framework in which knowledge can be encoded into rules in a simple *if (hypothesis) then (conclusion)* format. In addition, the Nexpert Development system provides the means for interacting with the user, directly executing external functions including C or FORTRAN subroutines, forward and backward chaining during a reasoning process, suggesting or volunteering data, and monitoring the execution of the system. In short, much of the functionality of the control executive discussed above is built into the Nexpert Development environment.

Unfortunately, Nexpert does not provide all of the functionality of the required control executive, nor does it provide the flexibility required to implement the necessary rules for evaluation of the knowledge system as designed under the Phase I effort. Specifically, Nexpert does not provide an *else* mechanism, and the knowledge designer cannot precisely direct the reasoning process to follow a set of steps by chaining to specific rules in the system. As a result, the Phase I demonstration system is limited in capability compared to the actual system that would be pursued in Phase II. Also, the use of the Nexpert environment required some relatively awkward work-arounds to create a feasibility demonstration for the Phase I effort.

The method by which Nexpert handles control is determined by the conclusions in the format *if (hypothesis) then (conclusion)*. In addition Nexpert provides the option to create rules using the format *if (hypothesis) then (conclusion) and do (procedures)*. In this way, when a rule is instantiated TRUE, a set of procedures can be executed, or variables can be assigned. It is by this mechanism, that the certainty factors for each rule are updated. Unfortunately, the procedures are not executed if the rule is instantiated FALSE, and chaining to another rule in particular cannot be indicated in the procedures. Thus, if conclusion B and its associated procedures must be executed if Rule A is false, other rules are required that contain all converses of A with the conclusion B. For example, to implement the following rule:

If (proc a is true) and
(proc b is false) and
(proc c is true) then
A is true and do x, y and z
Else
B is true and do u,v and w.

The following rules are required in Nexpert.

If (YES proc a) and (NO proc b) and (YES proc c) Then A and do (x, y, and z).
If (YES proc a) and (NO proc b) and (NO proc c) Then B and do (u, v, and w).
If (YES proc a) and (YES proc b) and (NO proc c) Then B and do (u, v, and w).
If (YES proc a) and (YES proc b) and (YES proc c) Then B and do (u, v, and w).
If (NO proc a) and (NO proc b) and (YES proc c) Then B and do (u, v, and w).
If (NO proc a) and (YES proc b) and (YES proc c) Then B and do (u, v, and w).
If (NO proc a) and (NO proc b) and (NO proc c) Then B and do (u, v, and w).
If (NO proc a) and (YES proc b) and (NO proc c) Then B and do (u, v, and w).

The only way chaining occurs in Nexpert is by including the *conclusions* of rules in the *hypotheses* of others. Therefore, to enforce a particular series of steps, each rule must check for a Boolean value indicating that the previous rule was executed. For example:

If (proc a is true) and (proc b is false) then C is true.
If (C is true) and (proc d is false) then D is true.

Thus, the second rule cannot be instantiated until the first is checked. By this method, we have implemented a particular set of steps for each test within the knowledge system. As a result, the knowledge system will dynamically execute image processing routines as necessary to determine the likelihood that a feature is of interest. But the knowledge processing will follow logical progression of tests when considering if the feature fits a certain description rather than chaining across different tests and unnecessarily executing image processing routines.

It is important to note, however, that if some data is *unknown* then its conclusion also will be unknown causing all conclusions in forward chaining to be unknown, as well. For example, in the 8-rule set above, if the results of *proc a* are inconclusive (ie neither TRUE nor FALSE), or *proc a* is not available, its value in Nexpert will become *NOTKNOWN*. When this occurs, all conclusions in the forward chaining direction that utilize the conclusion from *proc a* will also become *NOTKNOWN*. Since the knowledge system is organized into sets of tests that are executed in a predetermined order, one *NOTKNOWN* will result in an inconclusive result for an entire layer. This problem is another significant limitation of Nexpert for this application.

Due to all the limitations and difficulties with Nexpert, the design of the rule base did not include complete converses of each rule, as described in the 8 rule set above. Each rule was encoded so that only one measured quantity was checked. Specifically, a rule may contain several Boolean values of the form " *execute procedure A = TRUE*". Such a statement would produce a prompt to the operator of the form: "What is the value of execute procedure A?". This prompt indicated to the operator that the knowledge base required procedure A to be executed on the current candidate feature. After running the image processing procedure, the operator entered TRUE, and the knowledge base then prompted for a measured result from that procedure. Thus, only a partial converse of each rule was encoded. For example, the rule:

If (execute local contrast) and (contrast.sigma ≥ sigma_threshold) then grid_1 TRUE.

would have only the partial converse:

If (execute local contrast) and (contrast.sigma < sigma_threshold) then grid_1 FALSE.

This methodology implies that all image processing capabilities will be available for all cued features. Unfortunately, this is not always the case. Polarimetry procedures, for example, are only available with multi-polarized SAR imagery. As a result, special rules were implemented to handle situations that may occur in testing the prototype involving certain image processing functions. These functions include polarimetry, DEM analysis and region growing capabilities. The special rules simply provide more complete converses to handle the event of the hypotheses *eval_image_processing_routine* are set to FALSE. Note, this situation is due to a limitation in Nexpert, and will be compensated in Phase II by using a different development system.

In keeping with the proof of concept mission of the Phase I effort, the user acted as a procedural control executive, and Nexpert acted as the symbolic controller using the rule chaining methodologies described above. The operator evoked the knowledge processor for each candidate feature extracted by intermediate image processing steps. In Phase II the control executive will be automated, but for the prototype a human operator carried out the procedural steps. The user began by collecting a finite amount of data concerning the SAR situation and other data about the image to be investigated. Then, based on the input information, an initial image processing based segmentation was executed to produce a list of candidate segments to be investigated by the rule base. The operator then activated the rule base for each candidate segment in turn. Once the inference engine was started, it prompted the operator to run certain image processing sub-systems to validate or add data as necessary. The result of each activation of the inference engine was to assign a numerical score as to the likelihood of each candidate segment being part of a road, or competing feature.

3.3.2 Proposed Phase II Control Executive

For Phase II, an integrated image processing and expert system development environment is desirable. One result of such an implementation is reduction of the integration task required for a rule-based system to communicate automatically with the image processing environment. For Phase II we propose to use the KBVision System by Amerinex Artificial Intelligence, Inc. KBVision combines a comprehensive library of image processing and image understanding subroutines with higher level Lisp programming in a Unix[1], X-Window System[2] environment.

By using the Lisp programming language, we will be able to develop a custom inference engine

1. Unix is a trademark of AT&T Bell Laboratories.
2. The X-Window System is a trademark of Massachusetts Institute of Technology.

for the task of automated classification and change detection. Rather than trying to improvise with a rule environment suited to a different task, we will develop custom applications and a frame-based implementation of the classification knowledge for specific feature types. In addition, since this system will become a module of our hyper-spectral registration and change detection procedural workstation, additional rules will be developed that take advantage of hyper-spectral image data.

KBVision is a multi-layer tool consisting of: a set of conventional image processing tools, a library of higher level image understanding sub-systems, a relational feature database in which token attributes are defined by the user, and a Common Lisp programming environment. KBVision is an open system providing for user extension at all levels. In Phase II we will incorporate hyper-spectral registration and change detection algorithms, and add custom C code programs implementing new algorithms for sub-pixel registration. The Lisp programming environment has direct access to the feature database, and can execute any procedural subroutine in the system. Thus, the control interface between the image understanding, registration and change detection, and the expert system is invisible in KBVision.

The Phase II knowledge processing module will have an interactive multi-color graphical user interface. The frame-based expert system will operate in concert with the user, highlighting possible targets according to the likelihood derived from the encoded domain knowledge. Features will be marked by colored outlines for clear identification by the operator. On a Unix platform, the expert assistant will carry out multiple decision tasks simultaneously providing dynamic screen updating of the information as it is processed. Since Unix is a multi-tasking environment, the operator may monitor automated processing of the expert system, or pursue other tasks. At any time, the user will be able to intervene to correct or add data to the Lisp processor.

The architecture of the knowledge module will provide for expansion so that knowledge bases for various feature types can be added to increase the functionality of the system. The Phase II effort will concentrate on development of a knowledge base for identification and classification of roads in hyper-spectral imagery. This development will draw heavily on the Phase I rules for road-type features in SAR imagery. Additional rules will be added to take advantage of hyper-spectral data sources. Provision will be made to handle additional rule systems in Lisp for other feature types.

3.4 Scoring Function

During the reasoning process there is a requirement to keep track of the outcome of each rule with respect to its importance to probability of interest. The issues and methodology explored during the Phase I effort are discussed in section 3.4.1, and the actual method used in the prototype is discussed in section 3.4.2.

3.4.1 Issues and Methodology

This section addresses the problem of how to systematically handle the uncertainties which occur during the rule-based processing described above. One source of uncertainty is that in practice such processing does not work with perfect inputs. Another is that most rules have exceptions, and it is often impractical to enumerate all exceptional circumstances. Therefore, the conclusions of rules often are couched in probabilistic or "probability-like" terms.

Incremental, stage-wise computations require independence among the predicates being combined to allow straight-forward probabilistic computation. Unfortunately, this is rarely the case unless the rules are specially organized into a tree structure. Therefore, Bayesian updating is often abandoned by many AI systems in favor of heuristic, but more computationally feasible constructs such as *certainty factors*. These concepts and their modifications will be discussed below.

3.4.1.1 Taxonomy of Methods

One taxonomy for the various methods of handling uncertainties divides them into ***intensional*** and ***extensional*** systems. Extensional systems are usually production systems, ie. rule-based, or procedure-based. Intensional systems are more declarative-based systems.

In an extensional system, uncertainty is represented by heuristically derived "truth values". The formula for uncertainty is given as some heuristic function of sub-formulas. An example are the *certainty factors* used in MYCIN [Shortliffe, 76].

Intensional systems consider uncertainty more rigorously. A calculus of subsets of possibility, as in the usual theory of probability or generalizations such as *Dempster-Schafer* or *Fuzzy Logic*, is the model in that case.

The trade-off between the two formulations is that extensional systems are computationally convenient but can be semantically sloppy. Intensional systems, on the other hand, are semantically clear but can be computationally difficult and awkward.

However, this classification is somewhat misleading since rules can be implemented in either way, with differing interpretations of a rule's strength, ie. $A \rightarrow B$ with strength m. In the extensional scenario, the conclusion of a rule, B, has its certainty factor updated by a function of the rule strength, m.. Here, rules should be interpreted as the summary of past performance by agents or experts. The conclusions are reactions to problem situations or evidence.

On the other hand, intensional systems interpret rules not as past decisions but as factual constraints. Two typical ways of handling uncertainty in this context are *Dempster-Schafer* and *Bayesian*. The latter interprets m as a conditional probability of B given A, ie. $p(B|A)$. Dempster-Schafer, on the other hand, does not make quite such a strong statement, but instead only asserts that the proposition [A and $\neg$B] is possible but is excluded with probability m.

3.4.1.2 Extensional Systems

Rule-based systems exploit the property of *modularity,* which consists of the desirable properties of *locality* and *detachment.* The former refers to the capability to force conclusions based on the rules at hand, regardless of other information. This is distinct from probability theory. For example, p(B|A) potentially changes once K is known, ie. p(B|A) must be replaced by p(B|A,K) unless there is a verification of the irrelevancy of K. *Detachment* refers to the independence of a conclusion from how it was derived.

The deficiencies of extensional systems stem from an inability to mirror some of the more subtle forms of plausible reasoning. One such missing item is *bidirectional inference.* A subtle occurrence of this phenomena in human discourse is *abduction*.. For example, if A→ B and B is subsequently known to be true, then A is somewhat more credible. In a rule-based system, this would lead to A→ B, followed by B→ A, ...etc.

Another example from discourse is *explaining away.*, ie. A→ B and C→ B, and later the fact that C is found to be true makes A less credible. To avoid such cycles, rule-based systems simply prohibit bidirectional inferences.

The ability to retract conclusions is also missing from extensional systems. If A→ B and A is thought to be "true" enough to instantiate B with some measure of credibility, then no subsequent evidence can undo this conclusion.

Finally, there is an inability to properly sort out correlated sources of evidence. For example, if A→ B, C→ B, and both A and C are found to have some measure of credibility then the conclusion for B will increase even if A and C stem from the same cause.

The following are examples related to the present surveillance problem which illustrate some of the more subtle deficiencies of extensional systems:

Example #1 (Limits of Modularity): The following example illustrates that plausible reasoning cannot always ignore the rest of the body of rules and proceed using only locally available rules. Those rules which can potentially change the conclusions are sometimes called *suppressors*, or *competitors* for interpretation in our present scheme for surveillance from SAR imagery.

Suppose that an image contour is smooth and corresponds to another similar contour, with the width between them about equal to the width of a multiple-lane road. However, a decision must be deferred since it is possible to obtain further measurements using double-bounce polarimetry, supporting the possible existence of a canal or forest line.

The nearby existence of a canal or forest line neither confirms nor eliminates the possibility of a road at the site in question. However, the nearby existence of trees, if confirmed, should somewhat undercut the road hypothesis.

Example #2 Bidirectional Inference): These examples illustrate that the uses of predictive and diagnostic information can be important, while their misuse can cause problems.

Abductive reasoning is a normal part of discourse: if A→ B, then finding B = true makes A more credible. Humans are able to do this both ways: the presence of a road would imply the physical alteration of the surround, and the existence of the second makes the first seem more credible, though certainly not conclusive.

However, extensional systems require the explicit statement of the latter implication and removal of the former to avoid cycling.

MYCIN permits only diagnostic reasoning and no predictive inferences. However, removal of prediction prevents *explaining away,* another subtle form of discourse: A→ B, C→ B, B and C are found to be true, making A less credible.

For example, finding evidence that an irrigation canal contributed to the alteration of a physical surround makes the existence of a road less credible, though not conclusive since the two could coexist.

Another situation is that of an antecedent becoming more credible and thereby causing the consequent to become less credible. For example, suppose a very noisy, and thus somewhat ambiguous, double-bounce polarimetry test indicates a possible boundary between single and double-bounce returns in a region which is known from previously available thematic maps to contain a forest. The polarimetric boundary may be a forest line. However, further processing indicates that in an adjacent region in the image this same noisy test result was triggered by a crop boundary, whose existence was confirmed by other more conclusive tests. There is now more credibility in the existence of a polarimetric boundary, but less credibility in the existence of the forest line because of an alternative case nearby.

Example #3 (Multiple/Correlated Sources of Evidence): Normally, multiple sources should corroborate conclusions and increase credibility of the conclusion. However, discovery of a common origin does not justify an increase in credibility.

For example, a number of rules of classification using polarimetric data depend on statistics of the parameters of the polarization ellipse. Suppose each one of these indicates the possibility of a certain region being agricultural crops as opposed to scrub vegetation. Care must be taken to avoid having these rules reinforce each other since they all stem from the same derived polarimetric parameters.

Many remedies have been proposed for these deficiencies, especially the problem of correlated sources of evidence. These include bounds propagation and user-specified combination functions.

User-specified combination functions, such as *certainty factors,* represent an attempt to supply an unambiguous rule for combination which substitutes for the lack of conditional probability information.

Because most correlations are unknown, certainty factors are usually combined under the assumption of high or low correlation. This results in upper and lower bounds which are input for later update computations. Unfortunately, this usually results in the unchecked expansion of such bounds into un-meaningfully large intervals.

More subtle problems, usually ignored, include higher-order dependencies which may be needed beyond pair-wise correlations. Finally, the subsequent occurrence of new evidence may itself dynamically create or destroy dependencies. The application of certainty factors to the Phase I prototype rule system is discussed in section 3.4.2.

Example #4 (Monotonicity): A more fundamental problem with rule-based logic is its *monotonicity* , ie. the inability to reverse degrees of belief. One example of this dilemma is the tension between property inheritance for subclasses and subclass specificity. For example, "urban roads are smooth", "smooth roads exhibit specular reflectivity", and "an urban road undergoing refinishing is an urban road" would lead to the conclusion that "an urban road undergoing refinishing exhibits specularity". However, such refinishing usually creates surface roughness, as a by-product of the removal of the upper layers from grinding, on the order of SAR wavelengths. Therefore, such a subclass of roads does not necessarily inherit the specularity property of the parent class, and subclass specificity should dominate.

Example #5 (Predictions triggering explanations): This example concerns problems with causal reasoning which occur when predictions trigger explanations. For example, a road leading into an agricultural field suggests that a strong edge signature will be present. The presence of a strong edge in an agricultural region suggests a possible irrigation canal. However, the presence of a road should not suggest an irrigation canal.

This monotonicity problem cannot be treated by the addition of probabilistic grey-values interpolating between *true* and *false*. The real source of the problem lies in the *transitivity* property of rules, ie. $A \rightarrow B$, $B \rightarrow C$, therefore $A \rightarrow C$.

Because of such difficulties, research has been directed toward new interpretations of the "if-then" paradigm. One recent example is *circumscription* [McCarthy, 86]. This system uses only probabilities which are either infinitesimal, 1 minus an infinitesimal quantity, or an intermediate value. Such an approach to logic seems very interesting and may yet prove promising in applications.

3.4.1.3 Intensional Systems

Belief networks are intensional systems which are practical and workable. Relevant facts are encoded as neighboring nodes in a graph. In this way, facts or information that need to be ignored can easily be recognized as such since they are not represented by neighboring nodes. On the other hand, relevant facts or information are immediately available. Examples used in practice include: *frames, scripts, causal chains, inheritance hierarchies.*

These structures are used in practice, although "pure" logicians don't consider them general enough. However, such structures allow the focus on a narrow problem domain which represents the only "successful" approach to AI so far.

Another workable, intensional system type is *graphoids*. The advantages of this graph-oriented structure include:

- irrelevant facts don't alter relevant relationships,
- compatibility with probabilistic dependencies,

A possible disadvantage is that such a structure requires causal, ie. recursive, construction.

Probability theory, including *Bayesian* updating, as a representation and calculus of uncertainty has its adherents and detractors in the field of AI. This has been an area of great controversy. Some of the disadvantages claimed by detractors include:

- "epistemologically inadequate" [McCarthy et al, 69],
- requires full knowledge of all relevant distributions,
- humans are notoriously bad estimators of probabilities,
- requires knowledge of relevancies/irrelevancies when updating,

However probability theory has many strengths, including:

- rigorous and well-understood,
- not just a calculus but represents "structure of reasoning",
- can process belief measures using contextual information,
- context dependent information can be represented by graphs,
- probabilities on graph structures can be changed by local propagation strategies.

The primitive concepts of probability theory which support structured reasoning are:

- *likelihood*
    - $L(e|H)=\frac{P(e|H)}{P(e|\neg H)}$

  is *likelihood ratio*,
    - formalizes the notion of one event being more likely than another,
- *conditioning*
    - $P(a|c) = \frac{P(a,c)}{P(c)}$
    - which allows:
        - non-monotonic reasoning,
        - formulation of degree of beliefs via context,
- *relevance* which allows change of belief guided by changes in knowledge,
- *causation* which supports notions of:
    - non-transitivity,
    - induced dependencies,
    - asymmetry to identify causal directionalities in non-temporal data.

At this point, generalizations on the relationship of some systems to probability theory are interesting. It is a theorem [Heckerman, 86] that any system which updates certainty weights in a modular, consistent fashion has a probability interpretation, ie. the certainty update function is a function of the likelihood vector L(e|H) given above. For example, the certainty factor, CF, in MYCIN, can be given as:

$$CF = \frac{L(e|H) - 1}{L(e|H) + 1}$$

Moreover, any system of such rules producing coherent rules must be equivalent to a directed tree. In this case, no two rules stem from the same premise.

A prominent, though controversial, non-Bayesian method for the representation and updating of uncertainties is the *Dempster-Shafer* (D-S) calculus. It represents an attempt to construct a model for computing with uncertainty without requiring specification of all conditional probabilities, which is one of the major difficulties with using the Bayesian approach.

The D-S formulation handles partially specified probabilistic models, and does not need to complete the model. Instead it is able to compute with the evidence available, generally keeping some evidential weight uncommitted.

Briefly, let m( ) be a *basic probability assignment (bpa)* defined on the subsets of θ, called the *frame of discernment*. Then:

$$m: 2^{\theta} \rightarrow [0,1]$$

m(A) = strength of argument supporting proposition A,

$$0 \leq m(A) \leq 1,\ m(\emptyset) = 0,$$

$$\sum_{x \subseteq \theta} m(x) = 1$$

A is called the *focal element*. Define a *belief* function, Bel(x), over θ by:

$$Bel(x) = \sum_{y \subseteq x} m(y)$$

Analogously, define:

$$P(x) = 1 - Bel(\neg x),$$

called the *plausibility* of x, is a measure of the probability that x cannot be disproven.

The interval [P(x)-Bel(x)], called the *belief interval*, represents the probability that both x and $\neg x$ are compatible with the available evidence. Unfortunately, this interval is often misinterpreted as a measure of ignorance or uncertainty.

Dempster's Rule of Combination forms the basis for combining bpa's:

$$m_1 \oplus m_2 (x) = K \{ \sum m_1(x_i)m_2(x_j) \}$$

where the sum is taken over all subsets $x_i \cap y_j = x$, and

$$K^{-1} = 1 - \sum m_1(x_i)m_2(x_j)$$

where the sum is taken over all $x_i \cap y_j = \emptyset$.

The normalization factor K is introduced to retain the probabilistic aspect. It really represents the total time spent in no-conflict with the given constraint x. However, this normalization is also the source of certain problems to be discussed below.

Unfortunately, the combinatorics of the Dempster rule of computation can be exponentially complex. Some recent research has been concerned with computationally tractable methods[Gordon et al, 85], [Shafer et al, 87]

Dempster's rule of combination leads to a hyperbolic shape for the graph of accumulated bpa vs. number of sources of evidence accumulated, ie. the D-S method is a *monotonic* function of the evidence presented. In particular, this means that if any piece of evidence ever imparts all of its strength to a proposition and its negation, the belief interval will subsequently remain zero, regardless of any further evidence.

In [Safranek et al, 90], a method is described which introduces constant parameters which control the saturation effects of this monotonicity. They also provide a mapping which relates confidence factors of image measurements to bpa's.

Because D-S only works with partial information, it generally provides only partial answers, and some probabilistic queries cannot be answered. For example, it does not estimate the probability that a given proposition is true. Instead, it essentially measures how close does the available evidence force a proposition to be true, ie. the probability of *provability.*

However, this generally allows a proposition and its negation to be simultaneously true for some percentage of the time. Also, D-S cannot incorporate conditional probabilities unless a complete probabilistic model is created.

The D-S method has numerous other interpretations for how it assigns values to logical propositions, including *message codes* [Dempster, 68], *multiple opinions of experts* [Hummel et al, 88], *deductive databases* [Zadeh, 86], as well as *probability of provability* [Pearl, 88]. All of these notions are distinct from the *measure of truth or belief*, or *likelihood* interpretation usually used in Bayesian probability theory.

Available prior probabilities can be interpreted as *random switch* phenomena. Such a formulation creates a separation between object level reasoning and meta reasoning, In [Pearl, 88] this boundary is modeled by this external switch which imposes truth values for a proportion of the time equal to the given prior probability. Unfortunately, conditional probabilities cannot be modeled by this switch.

Therefore, since D-S operates not just on an object level as does Bayesian reasoning, it represents hypothetical envisioning. But *provabilty* rather than *likelihood* is the basis for formulating questions in this hypothetical setting.

For real problems, the notion of provability may seem artificial. However, hypothetical questions about possible scheduling situations seem especially well-suited to this formulation. Such questions inherently seem to involve two levels of knowledge - the compatibility constraints and the available prior probabilities. The compatibility constraints generate object-level statements which are assigned probability measures.

It may seem paradoxical that Bayesian probability, which works with more complete information than does D-S, does not seem suited to such queries. However, this is because probability theory is formulated only on a single object-level, while D-S requires the compatibility constraints to be outside the probabilistic model.

The D-S method also suffers from monotonicity, and does not conveniently allow the retraction of previously held beliefs given new evidence. This problem is a result of the normalization factor described above in the Dempster rule of combination.

Another problem is that in certain cases D-S overweights prior probabilities compared to the Bayesian formulation according to [Yen, 89]. Likewise, [Pearl, 88] gives an example using the famous "Three Prisoners Problem" where D-S gives extreme probability values compared to the Bayesian approach.

In [Hummel et al, 88], the *voting model of experts* allows a Bayesian interpretation which allows incorporation of covariance data. This interpretation and its connection with the notion of *provability*. deserves further research.

In conclusion, one may say that D-S represents an interesting attempt to circumvent some of the perceived difficulties with Bayesian probability (see [Cheeseman, 85] for a spirited defense of probability). As discussed above, D-S has numerous conceptual and computational difficulties, and such is not recommended in its present form for use in our system. However, it may become useful in the future especially for scheduling problems and other inherently "multiple-level" problem formulations.

3.4.2 Algorithm Used for the Phase I Prototype

As the rules are fired, and the system progresses through a reasoning session, the accumulation of knowledge concerning a particular feature must be stored in the system. A numerical scoring function provides a probability-like mechanism for the user to evaluate at the end of a session. The output then, indicates that a cued feature may be the feature of interest, such as a road, with certainty factor N%, and it may be another competing feature with a certainty of M%. N and M would be large and small percentages, ideally. This method enables the system to produce results that are non-Boolean, giving the user more of a feeling for the dependability of the solution.

A numerically based scoring system must be defined by five important requirements. These requirements are:

1. Given some initial certainty, p_0, for a particular feature, each rule with a TRUE result should increase p and each FALSE result should decrease p.
2. Successive TRUE results should monotonically increase p.
3. When there is a preferred ordering of the tests, earlier tests should have a more profound effect on the final score, p_N, than later ones. When there is no preferred order, the order of execution of rules should not impact the final score.
4. With no initial data, the initial certainty, p_o, for all features should be set to 0.5
5. The certainty factor for a TRUE result should be greater than 0.5 while the certainty for a FALSE should be less than 0.5.

We can assign a certainty factor, p_C, to each rule such that if the rule is instantiated (TRUE) then the cued feature is a feature of interest with certainty p_C. For example:

if (feature) is linear then it is a road with certainty p(road) = 0.8 else p(road) = 0.2.

This is adequate except that all certainties are not independent. Say for example that the given feature was linear, and highly contrasting with its background. However, its background consisted of a variety of circular objects. Clearly, the feature is contrasting with its background to some extent because it is linear. Thus the two conditions are not independent. As a result, conventional Bayesian probability techniques cannot be applied. For example:

if feature is linear then it is a road with probability p(road) = 0.8 else p(road) = 0.2.
if feature is contrasting then it is a road with probability p(road) = 0.6
else p(road) = 0.4.

Now assuming the initial probability p(road) = 0.5, and both rules are true, should the probability p(road) be 0.8, 0.6, more than 0.8, an average of 0.5, 0.6, and 0.8, or some other value. Being that the conditions are not independent, we cannot take each one as absolute and change the probability as in Bayesian systems. Averaging the probabilities, dynamically will lead to an increase, then a reduction in the probability. That is, the average of 0.5 and 0.8 = 0.65. Thus after the first rule, p(road) = 0.65. Then if the feature is contrasting, the value of p(road) will be updated to 0.633. We see that even though the result of the second rule was true, and the certainty factor was greater than 0.5, the actual probability, p(road) was decreased.

3.4.2.1 Order Dependent Score Function

One solution to these problems is to place the tests in a preferred order such that the most important test is first, followed by the next most important, etc. Then an order dependent scoring function can be used that consists of a weighted average of p_0, all previous certainties, and a derived certainty factor. The new certainty factor is derived from the nth iterated value of p, p_n, and the percentage above p_0 of the certainty factor p_C. The derived certainty factor is the sum of p_n and the computed percentage multiplied by the "room to grow (or shrink)" of p_n. See Figure 3.7 The order-dependent update rule is as follows:

$$p_{n+1} = (np_n + \text{abs}(\text{nint}(p_C) - p_n)p_a) / (n+1)$$
$$p_a = (p_C - p_0) / p_0$$

where:

- n = number of rules instantiated
- p_n = certainty of the current feature after the nth rule has been instantiated
- p_{n+1} = updated certainty after the current rule
- p_C = certainty factor of the current rule
- p_a = increase or decrease of current certainty factor based on p_C

and:

$0 \leq p_n \leq 1$,
$0 \leq p_{n+1} \leq 1$
$0 \leq p_C \leq 1$
$-1 \leq p_a < 1$

Note the system can be taken one step further utilizing previously obtained certainty factors to influence later ones. The certainties could then be carried through the system, for example:

if (avg curvature measurement ≤ 1.0) then p(linear) = 0.8 else p(linear) = 0.2.
if p(linear) > 0.75 then p(road) = (0.8+p(linear))/2
else if p(linear) < 0.25 then p(road) = (0.2+p(linear))/2.

This rule provides a gray area between definitely linear and definitely nonlinear. If the linearity falls between the two thresholds, p(road) is not affected at all. Similarly, values returned from image processing subroutines, such as an intensity threshold to test for terrain effects, can be used to directly impact the certainty factors for that rule. For example:

if (percentage of pixels above threshold, p_a > 50%) then p(terrain effect) = p_a/100
else p(terrain effect) = 0.5.

Note, this rule provides a variable certainty factor on TRUE, but will not affect the derived probability on FALSE.

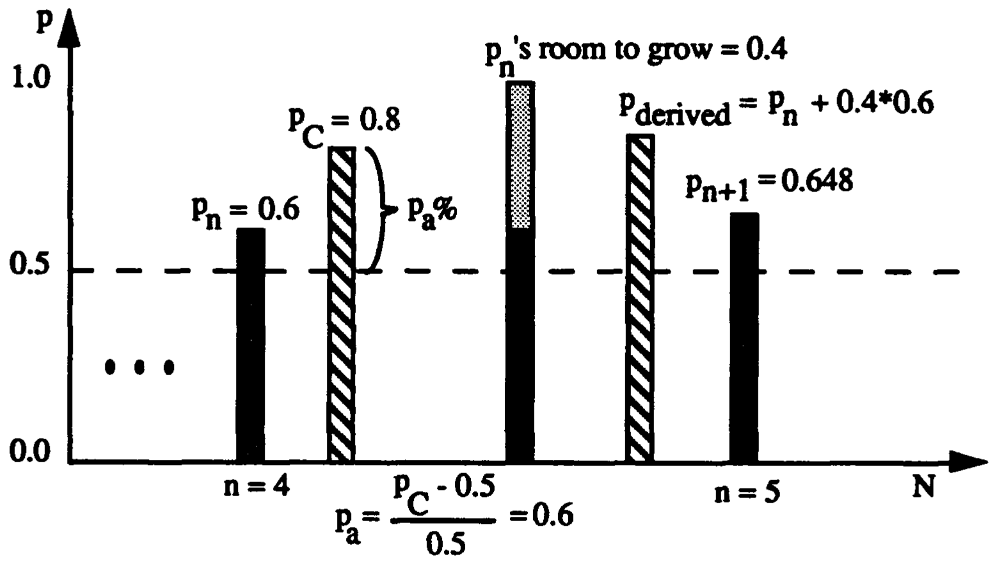


Figure 3.7 Graphical explanation of scoring function.

To better understand the effect of the scoring function on the final certainty factor given varying rule certainties, the function was tested under a variety of conditions. The first of the general requirements outlined above states that each TRUE result should increase the score and each false should decrease it. Thus, if a series of perfect certainty factors (1.0) were introduced, the function should increase monotonically. This behavior is illustrated in Figure 3.8. Similarly, a series of 0.0 certainty factors should cause the function to monotonically decrease. See Figure 3.9. The response of the scoring function to a random series of conditionals is shown in Figure 3.10. Figure 3.11 illustrates the order-dependence of the function. In Figures 3.7. through 3.16,, the solid black squares indicates the certainty factors for each rule, and the open squares indicates the updated factor.

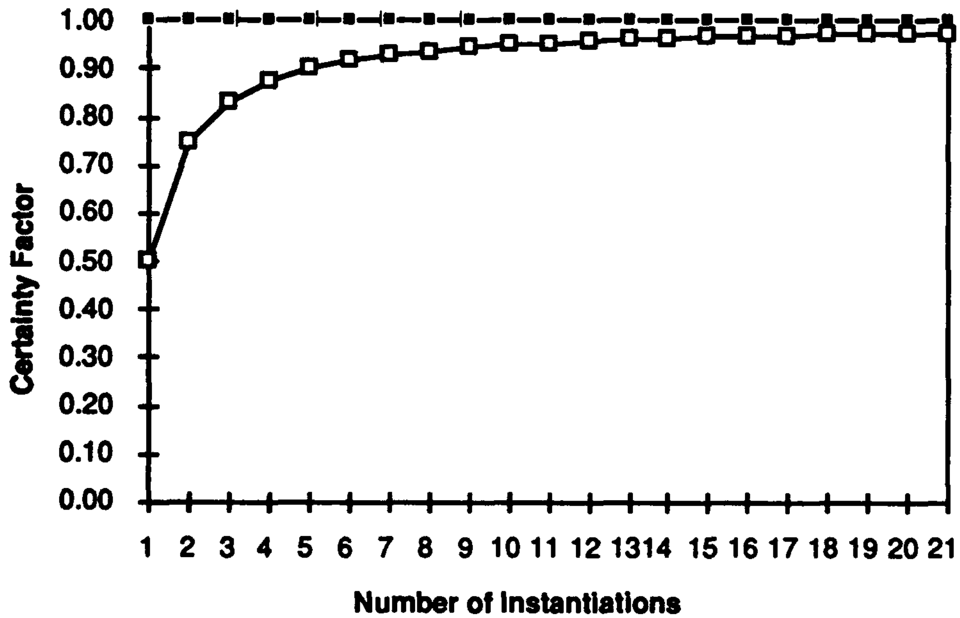


Figure 3.8. Behavior of order-dependent scoring function given perfect conditionals.

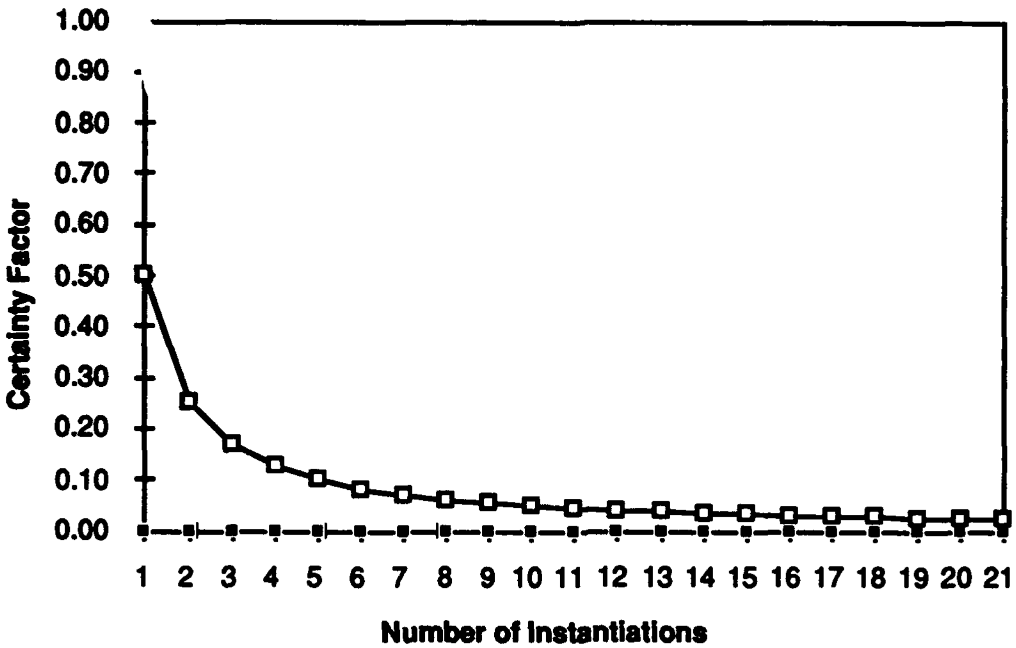


Figure 3.9 Behavior of the order-dependent scoring function given false conditionals.

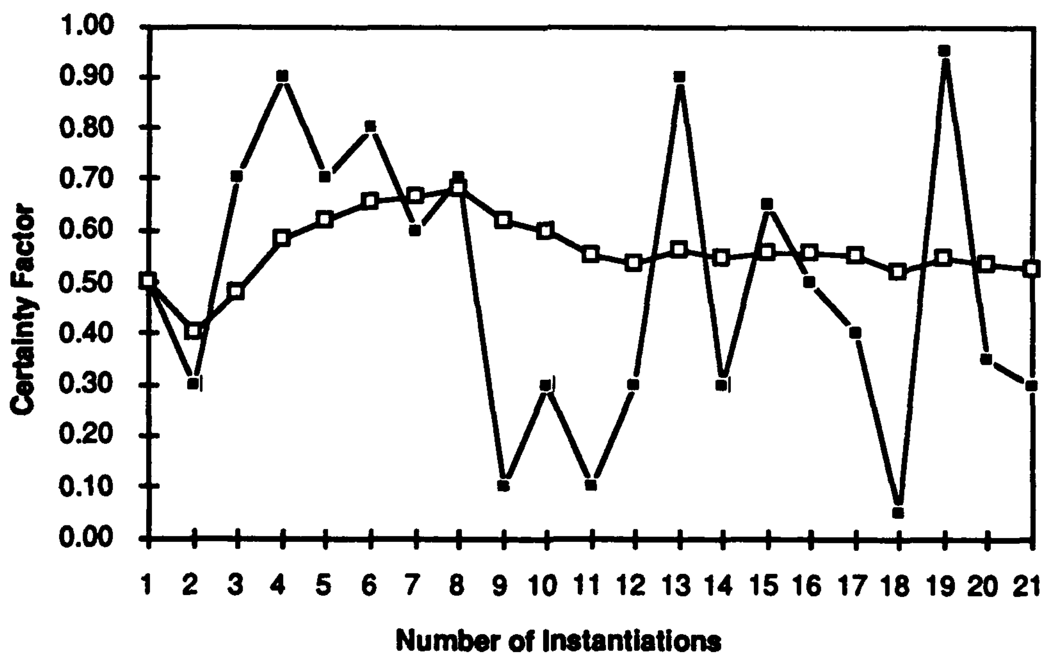


Figure 3.10 Behavior of order dependent scoring function given random conditionals..

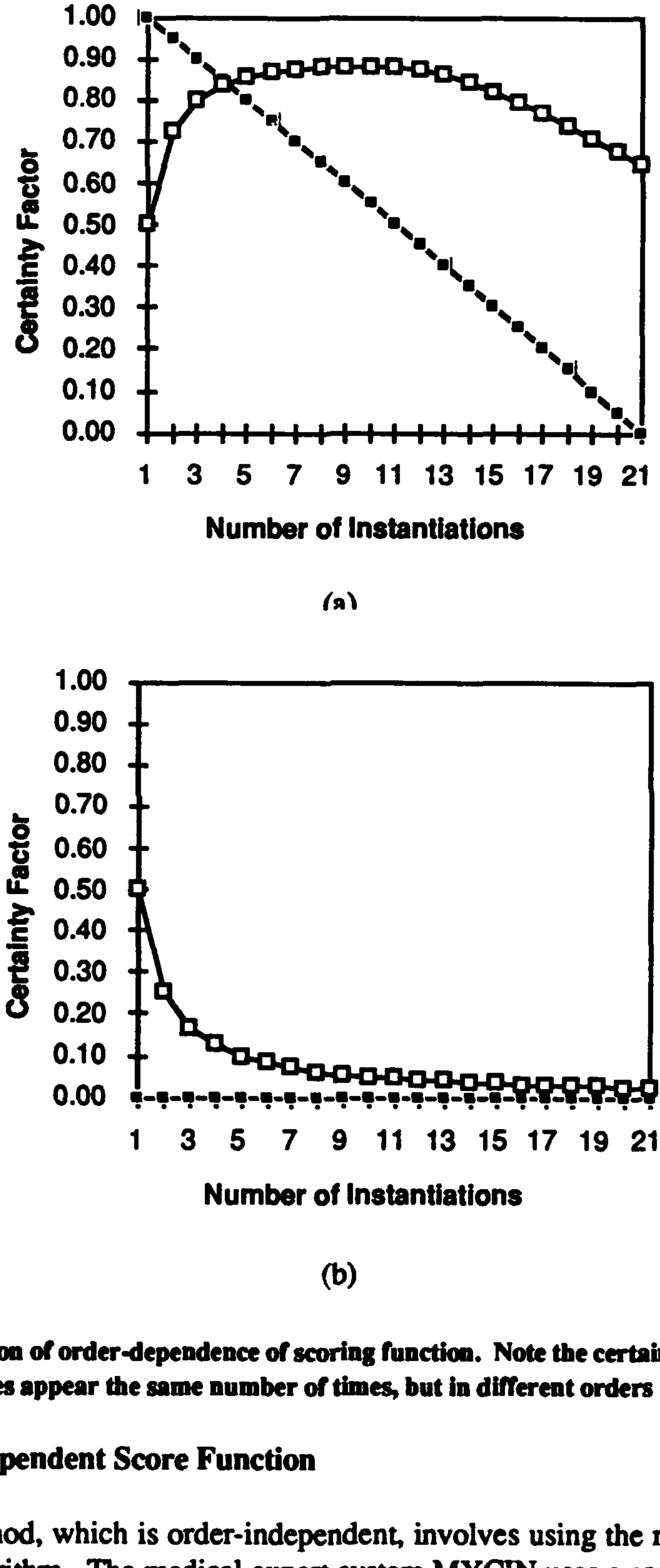


Figure 3.11 Illustration of order-dependence of scoring function. Note the certainty factors indicated by black squares appear the same number of times, but in different orders in (a) and (b).

3.4.2.2 Order Independent Score Function

Another scoring method, which is order-independent, involves using the rule certainty factors in a non-averaging algorithm. The medical expert system MYCIN uses a scoring algorithm similar to the one presented above, but without weighting rules according to order.

Recall from section 3.4.1.1, order-independent scoring functions based on certainty factors represent an attempt to supply an unambiguous rule for combination which substitutes for the lack of conditional probability information. This usually results in the unchecked expansion of such bounds into un-meaningfully large intervals.

We illustrate some mathematical difficulties with the usual definition of certainty factors given in MYCIN [Shortliffe, 76]:

Given two certainty factors p_n, p_a, the new, updated certainty is given by the rule:

$$p_{n+1} =: \begin{cases} p_n + (1 - p_n)p_a & p_n, p_a > 0 \\ p_n + (1 + p_n)p_a & p_n, p_a < 0 \\ \dfrac{(p_n+p_a)}{(1-\min(p_n,p_a))} & p_n p_a < 0 \end{cases}$$

Note: the range both of p_n and p_a in this derivation are [-1 , 1], rather than just p_a as in the previous section.

For example, consider an initial certainty factor $p_n = 0.6$, and a rule certainty factor, p_a of 0.8. As in the previous section the "room to grow" factor $(1-p_a) = 0.4$ is used in the update function. The updated certainty is computed by increasing the current factor by 40% of the update factor producing a new certainty of 0.92. Notice that if the order was reversed, and the initial factor a = 0.8 and b = 0.6 that 0.8 would be increased by 60% of 0.2 to produce 0.92. The update is similarly computed for negative certainty factors.

The problem with this formulation occurs for the case $p_n>0$, $p_a<0$. Suppose $p_a \rightarrow 0$. Then the denominator $(1-\min(p_n,p_a)) \rightarrow (1-p_a)$. Therefore, the quotient:

$$\frac{(p_n+p_a)}{(1-p_a)} \rightarrow \frac{p_n}{(1-p_a)}$$

This quantity is discontinuous with the branch for $p_n,p_a>0$. This results in the function being order *dependent* when jumping between branches, although it is order independent *within* branches. The discontinuity could be removed by making a new rule:

$$p_{n+1} =: \begin{cases} p_n + p_a - p_n p_a & p_n, p_a > 0 \\ p_n + p_a + p_n p_a & p_n, p_a < 0 \\ p_n + p_a & p_n p_a < 0 \end{cases}$$

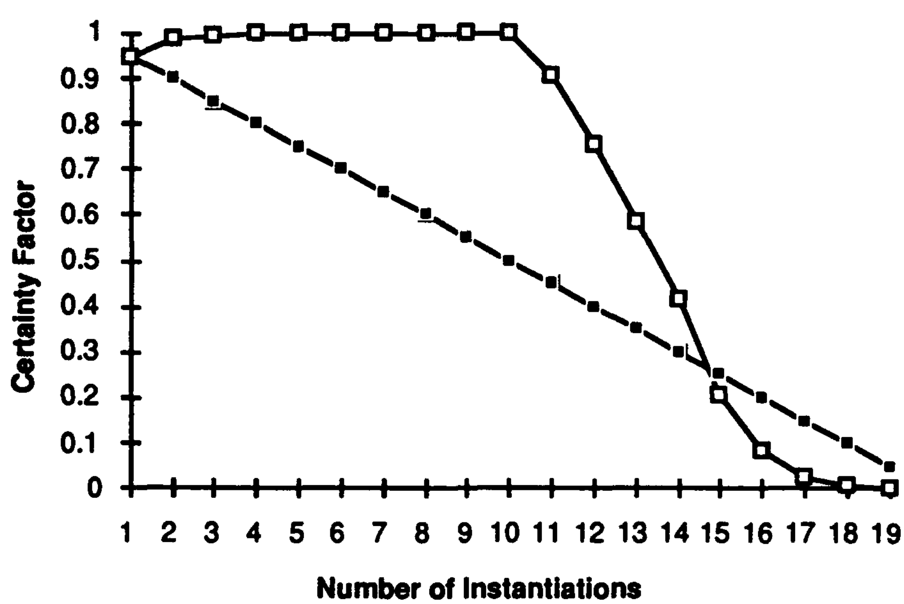
Now, the update function is continuous, and monotonicity holds for the two branches $p_n,p_a>0$ and $p_n,p_a<0$, and non-monotonic for the branch $p_n p_a<0$. However, it is still not order independent when jumping between branches. This can be remedied with a rule in which only two branches are used:

$$p_+ = p_+ + p_a - p_+p_a \qquad p_a > 0$$
$$p_- = p_- + p_a + p_-p_a \qquad p_a < 0$$
$$p_n = p_+ + p_-$$

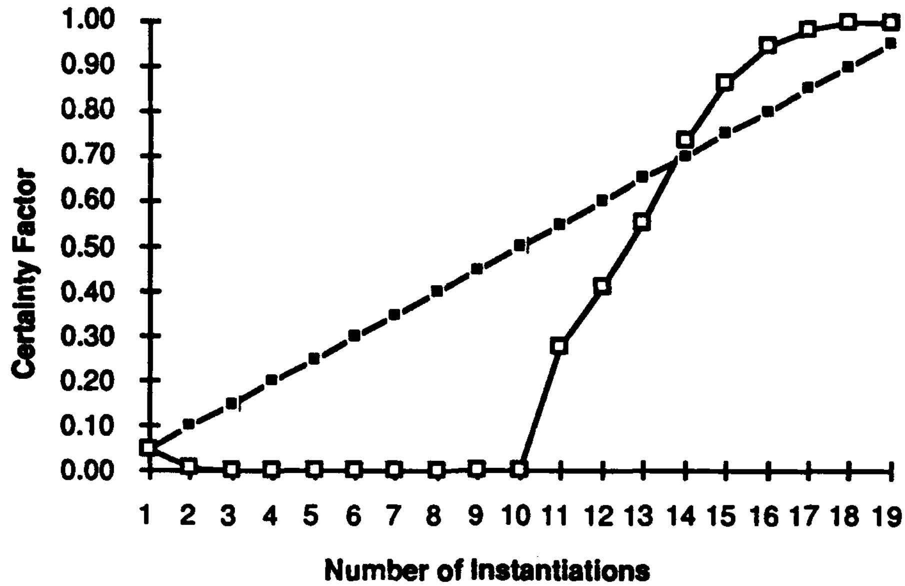
The case $p_n p_a<0$ is never used for permanent updating. Instead, each branch is *separately* updated. At any time, they can be added to get the present *total status*, p_n. However, any new data is updated using the correct branch, rather than the previous total status. Each branch is initialized to zero.

This function is now monotonic within each branch, non-monotonic for the case $p_n p_a<0$, continuous everywhere, and order independent. The latter property follows from the symmetry with respect to the variables p_n and p_a within each branch, and the fact that these branches are the only ones to be updated (separately).

In the following Figures 3.12-3.16 pn has been normalized to a range of $0 \leq p_n \leq 1$. Figure 3.12 illustrates the difficulty with the original MYCIN method. Figure 3.13 through 3.16 illustrate the new function described above. Figures 3.13-3.16 may be compared with 3.8-3.11. Note in particular Figure 3.15 in which a random sequence of conditionals produces a final certainty of 0.5. This is the expected result, and illustrates the order independence of the function.



(a)



(b)

Figure 3.12 The MYCIN scoring function is order independent in each branch, but the discontinuity in the cross-over function causes differing results given data in reverse order.

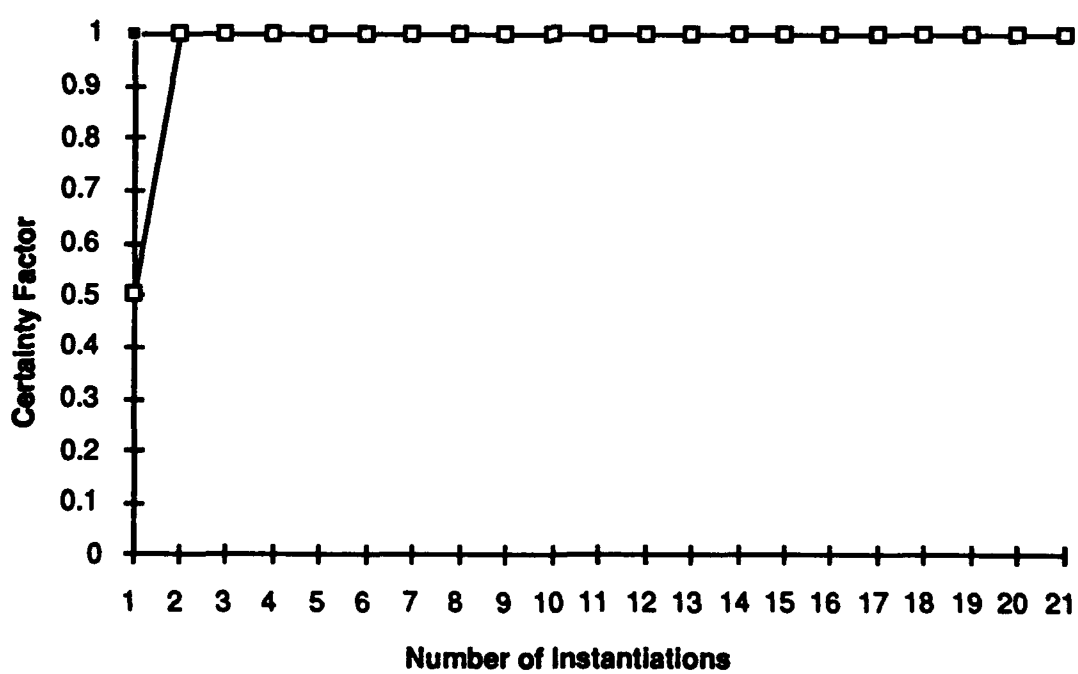


Figure 3.13 Behavior of the order-indpendent scoring function given TRUE conditionals.

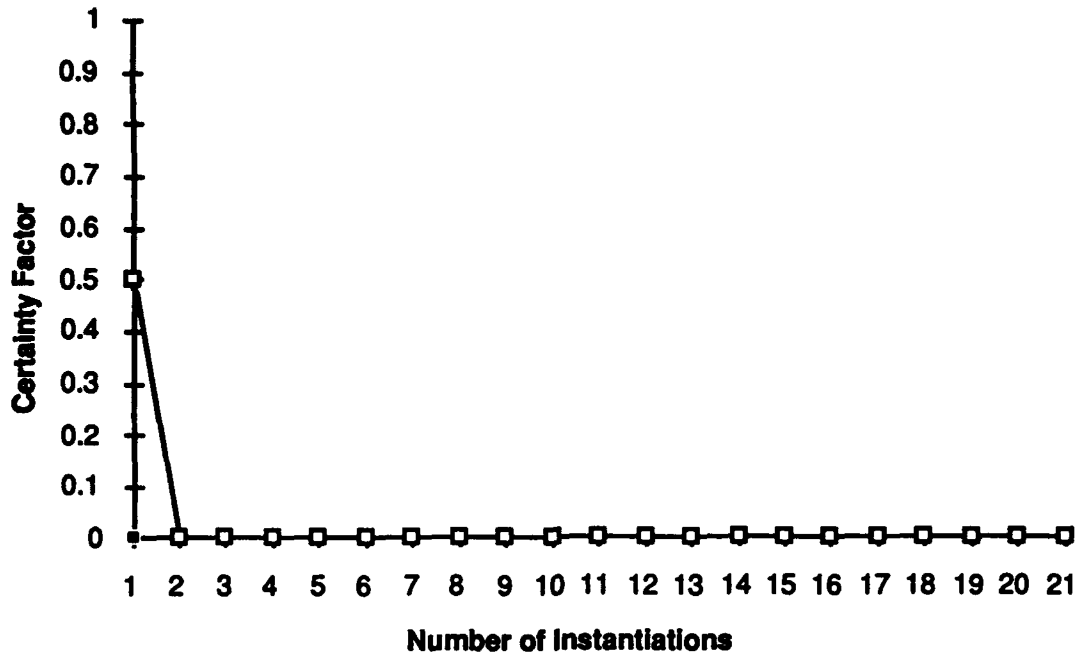


Figure 3.14 Behavior of the order-indpendent scoring function given FALSE conditionals.

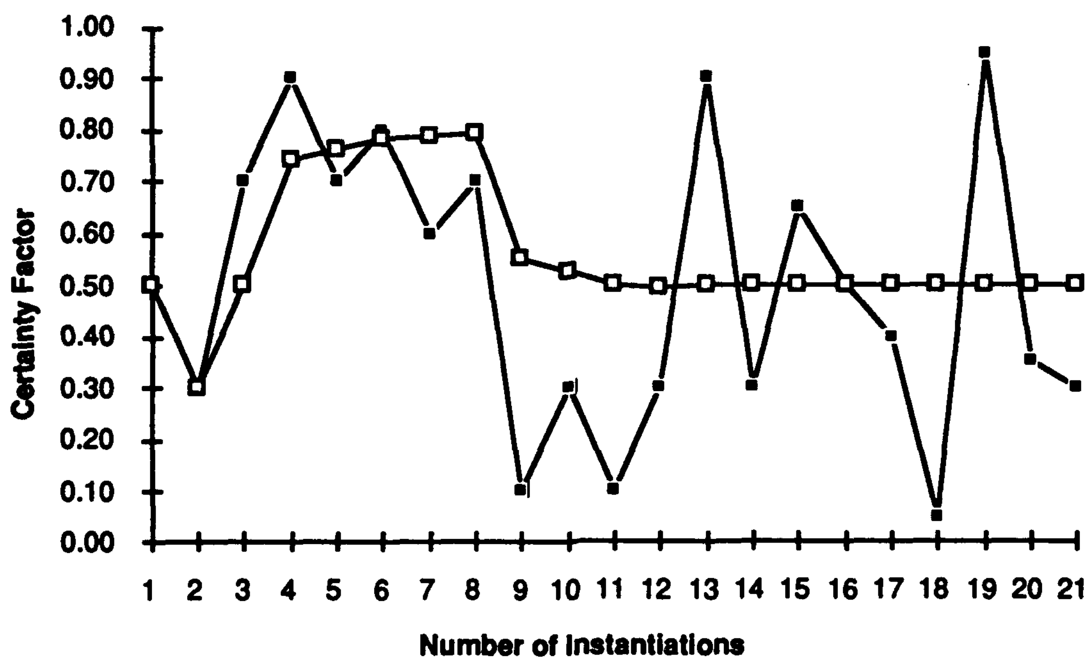
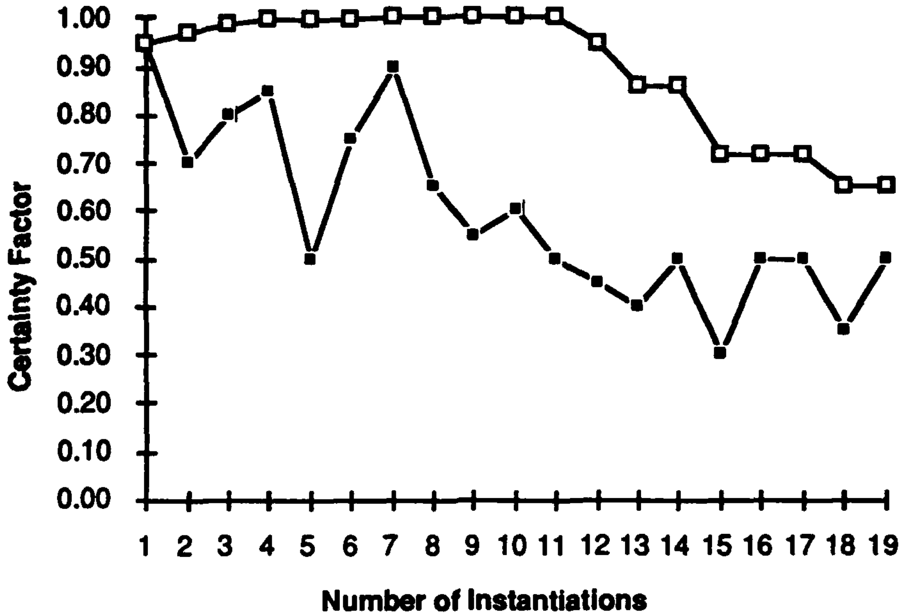
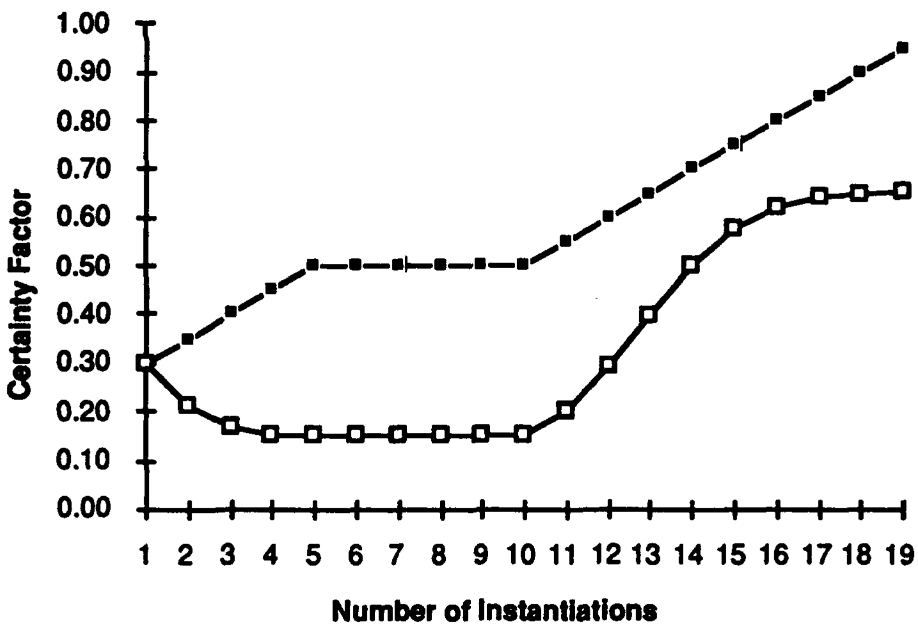


Figure 3.15 Behavior of the order-indpendent scoring function given random conditionals.
This Figure may be compared with Figure 3.10.



(a)



(b)

Figure 3.16 Illustration of order-independent scoring function. Note the rule certainty factors indicated by black squares appear the same number of times, but in different orders in (a) and (b). The final certainty factor in both cases is 0.65 arrived at in different order. The rule certainty factors are offset toward the positive to indicate that the function does not default to 0.5.

4.0 Image Processing Interfaces

A hybrid system in which both symbolically-based subsystems and procedurally-based subsystems are employed, either system can assume part of the duty of the other. As a result, the final product may have a very complex symbolic system linked to a very simple procedural system or vice-versa. In the Model-Based SAR Feature Analysis System, we have made an effort to divide the duty equally so that relatively simple image processing subroutines are triggered by an equally simple reasoning system.

The orthogonal breakdown of knowledge organization described in section 3.2 indicated a list of distinguishing characteristics for both features of interest, and competing features. These distinguishing characteristics must be determined by procedural image processing primitives. Recall Table 3.1 in which the distinguishing characteristics were listed across the top row. Many of these characteristics are related. For example, linear and curvilinear features can be identified by the same subroutine with the former being a special case of the latter. Section 4.1 briefly explains the organization of the image processing structure, and the algorithms used to determine characteristics. Section 4.2 describes the I/O requirements of the image processing routines.

4.1 Organization and Algorithms

The 28 distinguishing characteristics listed in section 3.2 were used as variables in the knowledge base to determine the likelihood of each cued feature being a feature of interest. The nature of the hybrid rule-image processing system provides the means for each of these characteristics to be determined procedurally using image processing subprograms.

The 28 characteristics fall into 11 categories: linearity, consistency, intensity, local contrast, local edge gradient, reflectivity, width, length, orientation, statistical distribution, and polarimetric effects.

There are 12 image processing primitives. Combinations of these primitives can be used to define characteristics in each category. The image processing primitives include: local tangent, thresholding, mathematical morphology, surface analysis of a digital elevation model, streak detection, edge operators, local lineal statistics, local regional statistics or local contrast, analysis of Mueller matrix, autocorrelation, and region growing. Table 4.1.1 shows the relationship between the image processing primitives and the feature characteristics. Some characteristics can be determined using several image processing primitives equally. Other characteristics require image processing to be used in combination to satisfactorily affirm or deny their values. Generally, one set of image processing routines can be used to differentiate the characteristics in one category. The following subsections describe the image processing algorithms and their applications.

4.1.1 Linearity

For example, the distinction between linear and curvilinear features can be determined using a mathematical morphology processor. In addition, the degree of linearity can be determined using various operators within the mathematical morphology sub-system. Another method for determining the degree of linearity of a feature uses the variance of local tangent at each pixel along a feature. A small variance implies a very linear feature, a large variance implies a curvilinear feature.

4.1.2 Consistency and Roughness

The consistency of a road feature, ie gravel, asphalt, concrete, etc, can be determined by the roughness of the surface. A gravel road, for example, is rougher than a concrete road. Radiation incident on a surface is reflected specularly or diffusely depending on the roughness of the surface. The Rayleigh criterion is most often used to specify the roughness of a surface. Consider a rough surface with height variations Δh, and radar look angle ϕ. The path difference between parallel phase fronts due to the height change is given by:

$$d = 2\Delta h \cos\phi$$

Leading to a phase difference of:

$$\Delta\Psi = 2\pi d / \lambda \;=\; 4\pi\Delta h \cos\phi / \lambda.$$

Note that if the phase difference is 0, then the surface is equivalent to a perfectly smooth one, but as the phase difference approaches π, the two phase fronts destructively interfere. Rayleigh arbitrarily chose the midpoint $\pi/2$ as the dividing line for roughness. Height variations leading to a phase difference of less that $\pi/2$ are due to a smooth surface, and other variations leading to a phase difference in excess of $\pi/2$ are due to a rough surface [Colwell et al, 83]. Thus if the height variations Δh are greater than $\lambda/(8\cos\phi)$ then the surface is rough, otherwise it is smooth.

When radar is incident on a smooth surface, the reflected wave acts as if it has bounced off a mirror, and produces very little reflected energy in the direction of the sensor. Thus a smooth surface will provide a specular reflection for the radar, resulting in a dark region in the image. A rough surface, on the other hand, such as a gravel road, or rock strewn field, will provide ample surface for radiation to be reflected back toward the imaging sensor. The resulting image will have a much more intense response from the rougher surface. An image processing system that uses a simple threshold, or a local regional thresholding algorithm based on local statistics can be used to distinguish rough surfaces from surrounding ones. Thus both intensity and local contrast are used to distinguish road consistency.

The water content of one region can be distinguished from another using one of two methods. First, two neighboring regions consisting of the same types of scatterers but differing water contents are likely to produce differing radar signatures due to the variation in the σ_0 value of the scatterers. This results in variation in the intensity of the scattered wave. Thus a local edge gradient operator can be used to distinguish regions of varying water content. Also, polarimetric responses differ depending on water content and relative dielectric constant. The number of bounces metric proposed by [Van Zyl, 89] can be used to infer variations in surface cover and dielectric constant.

4.1.3 Statistical Distribution

The streak detection image processing primitive can be used to discriminate a variety of features that are characterized by long narrow curvilinear signatures. Streak detection uses the statistical variation of local summed profiles to find linear signatures that contrast their surroundings. Streak detection will segment only those signatures that are roughly one to two pixels wide, and have gray values sufficiently disparate from the surrounding region. As a result streak detection is ideal for identifying narrow roads of various materials when the surrounding surface material is different. Also, because of the unique linked list data structure in which the signatures are stored, streak detection provides a simple means for determining length of a cued linear feature. Similarly streak detection can also be used to determine the number of lines within a region, and when used in combination with the local tangent operator, can determine the number of roughly parallel lines.

4.1.4 Surface Facet Analysis

The use of a digital elevation model (DEM) is required if the knowledge base is to use motion of scatterers as a detection mechanism. That is, the motion of a river, or water in a canal or exposed aqueduct can be used to differentiate the feature from a road or tree line. The change in position of the water during the imaging period will introduce an extra Doppler shift to the pixels in which the water was imaged. When the radar is correlated, creating an image from the raw phase history, the residual motion of the water will cause the water pixels to be displaced from the river bed or canal. Using a DEM, the location of the river bed can be determined by finding the local minima in the surface. Given that the SAR image is registered to the DEM, a river may be distinguished if the pixels corresponding to the water are displaced from the location of the minima in the DEM.

A DEM can also be used to distinguish a terrain highlighting effects caused by foreshortening from true features of interest. If the cued feature consistently lies along a ridge or edge of a plateau, or is consistently adjacent to a shadow in the down-range direction , the cued feature is likely a terrain induced highlighting effect.

4.1.5 Edge Detection and Local Statistics

Edge operators such a Sobel, Roberts [Ballard et al, 82] and Lee sigma edge detector can be used to segment regions of varying water content or scattering properties. Edge detectors can also be used to find systems of roads in clutter.

Once a linear feature has been detected, local lineal statistics can be used to determine the amount of variation in the intensity of the pixels along the cued segment. Roads that are parallel and equidistant to the flight path, in particular, will generally exhibit the same, or nearly the same pixel intensity all along their length. If the width of a cued segment is known, local lineal statistics can be used to estimate the likelihood of the segment being a road. If the width is not known, local lineal statistics can be used to determine a width given a standard deviation about the mean. Then if that width is within one of the discrete ranges of widths that roads are likely to fall into, the likelihood of the cued segment being a road can be increased.

Similarly, local regional statistics can be used to identify the borders of regions or to assign a likelihood that a region is of a certain type. For example, shadowed regions in SAR imagery usually exhibit a locally bimodal histogram [Curlander et al, 89]. Local statistics can determine if a bimodal histogram exists within a region, and delineate the borders of the region. Using this method alone, if a bimodal histogram is not found, the likelihood of a shadow being present in the region can be reduced, but another test is required if the likelihood of shadow is to be increased. By comparing the location with a co-registered DEM, the existence of a SAR shadow can be determined.

4.1.6 Autocorrelation

Another method for finding repeated patterns of lines, a characteristic of urban road grids, is autocorrelation. An image known to contain a grid in a noisy background can be correlated with itself, and the resulting image thresholded to produce the locations of the grid lines within the image. This method has been used successfully by Vexcel on other projects with sub-pixel accuracy.

4.1.7 Polarimetric Effects

When a polarimetric SAR sensor is used to create the image to be analyzed, the equivalent of a complex scattering matrix is stored for each pixel. This scattering matrix can be used to create images corresponding to any combination of transmitted and received polarization. This polarimetric information can be used to classify radar wave scattering interactions for various targets. Scattering classes such as exact number of reflections cannot be determined for a given pixel. But by examining the change direction of orientation of the scattered wave with respect to the transmitted wave, and the handedness of the scattered wave, the number of reflections can be narrowed to even or odd classes.

For example, it has been found that water surfaces exhibit a simple class of scattering characterized by an odd number of reflections [Van Zyl, 89], while scattering from an urban area is similar to that predicted by primarily even number of reflections. By creating binary images denoting which pixels fall into the even, odd and diffuse reflections class, credence can be lent to the status of a particular candidate feature. Also, since corner reflectors can be associated with the class of even number of reflections, a region containing a high density of corner reflectors can also be delineated. Agricultural fields have been found to exhibit scattering mostly similar to an odd number of reflections, while tree covered regions are characterized by a mixture pixels from even, odd and diffuse classes.

As a result, it is possible to create rules which utilize images created from various combinations of polarimetric data, to evaluate the likelihood of a candidate feature falling into a particular thematic class.

4.2 I/O requirements

Because of the non-Bayesian scoring system in which a certainty factor is associated with each rule, there is a requirement that each image processing subsystem return a value to the rule base indicating a numerical measure of the goodness of response of the cued feature. This measure should ideally be a non-binary value so that the conditional probabilities can reflect the indeterminant nature of evaluating the cued data. For example, the result of calling the Sobel edge detector for some region may include the locations of all edge gradients in the region, their orientations, and the strength of the gradient. The gradient strength can then be used to influence the certainty factor assigned to that rule.

In this section, the input and output requirements of each of the image processing primitives listed in section 4.2 (and shown in Table 4.1.1) are described. Each description will include the data to be passed from the knowledge base, through the control executive, to the image processing subroutine, and the data to be returned through the control executive to the knowledge base. The intricacies of how the data is used in the knowledge base will be covered in section 5.0 of this report.

<u>Analysis of Mueller matrix</u>

- Purpose: To delineate regions by even or odd bounce characteristic, or to provide data to analysis for variation in relative dielectric constant. Also, special masks can be used to compare the Mueller matrix at each pixel to the signature of some known material.
- Input: region of a multi-polarized image, indices of terms in matrix for special processing, key number indicating special processing
- Output: for each pixel: # of bounces, results of special mask processing

<u>Autocorrelation</u>

- Purpose: To detect repeated patterns
- Input: image or region of image
- Output: Boolean indicating if periodic pattern was detected
period (spacing between individuals) of the pattern

<u>Bimodal Histogram</u>

- Purpose: To determine if a region is characterized by a bimodal histogram, such as regions of SAR containing shadow, or water
- Input: image or region of an image, initial number of bins to divide up histogram
- Output: Boolean indicating presence of bimodal histogram, and threshold intensity

<u>DEM surface analysis</u>

- Purpose: To determine if a given feature lies along a drain line or in proximity to a ridge line in an image. This analysis is part of determining if a cued object is in motion, or a terrain induced shadow.
- Input: vectorized representation of a cued feature, registered DEM, key to procedure
- Output: Boolean indicating if cued features lie along a drain line, or near ridge

<u>Edge operators</u>

- Purpose: To determine the gradient value of pixels along a cued linear feature
- Input: vectorized representation of an object, or a region, key indicating type of input

Output: values of gradient at each input point, mean, variance of gradient values, orientation of edges

Local Lineal Statistics

Purpose: To determine the variation in brightness of a lineal object along its length

Input: vectorized representation of a lineal object, width (optional)

Output: mean, variance, max and min intensities along object

Local Regional Statistics (Local Contrast)

Purpose: To determine the variation in brightness of regional, or lineal parts of a region

Input: image or region of an image, number of standard deviations used to delineate regions

Output: vectorized representations of delineated regions, image containing pixels whose values are the number of standard deviations above the local means.

Local Tangent

Purpose: To determine the linearity of a cued feature, or to indicate the direction perpendicular to the cued feature at a given point.

Input: vectorized representation of feature, point location along feature

Output: mean, variance of turning angle of feature, direction perpendicular to given point

Mathematical Morphology

Purpose: To delineate locations of features with given shapes within a region

Input: Morphological mask, region of image

Output: vectorized representation of all features in region segmented by morphological mask

Periodic Highlights

Purpose: To determine if a cued linear feature exhibits periodic highlights, such as a transmission line.

Input: a vectorized representation of a linear object

Output: Boolean indicating if input linear object is characterized by periodic highlights

Rayleigh Histogram

Purpose: To determine if a region is characterized by a Rayleigh histogram implying that the pixel values are dominated by speckle noise.

Input: image or region of an image

Output: Boolean indicating presence of Rayleigh histogram, mean and variance according to Rayleigh distribution.

Region Growing

Purpose: To delineate regions with statistically different pixel values than their surrounding region, such as a park in the center of a cluttered city.

Input: image region containing a suspected contrasting area

Output: Boolean indicating if such a region has been found
vectorized outline of region

Streak detection

Purpose: To delineate linear features characterized by local contrast

Input: region suspected to contain such features

Output: vectorized representation of such features

Threshold

Purpose: To delineate regions or linear features whose pixel intensity is in excess of a threshold, or lies between a range of thresholds

Input: image of region of an image, minimum and maximum intensities indicating pixels to be turned on in output image

Output: binary image

Vectorization

Purpose: To extract vectorized representation of lineal or regional features from a binary image. The vectorized representation consists of an array containing the length, and all locations of pixels along an unbroken contour.

Input: binary image or region, minimum contour length to be saved

Output: vectorized representation of all contours in input region

Width Profiler

Purpose: To determine the width of a linear feature. This subroutine must use the local tangent subroutine initially to determine in which direction to measure width

Input: vectorized representation of a linear feature, a point along the feature, criteria used to determine width (eg intensity threshold).

Output: width of feature at given point, mean, variance of feature throughout its length

5.0 Description of Rules

Both the "meta-rules" and the encoded Nexpert Object 1.1 rules are discussed in this section including the architecture of the prototype knowledge base within Nexpert. As mentioned in Section 3, initial efforts during the Phase I project were concerned with developing rules according to areas of knowledge and properties. These rules and the reasoning or factual basis behind them is discussed in Section 5.1. Section 5.2 discusses the architecture of the prototype, and the methods by which the initial rules were incorporated into the prototype. The actual list of encoded Nexpert Object rules are given in the appendix along with a glossary of terminology and variable names.

5.1 Meta Rules

The following rules are broken down by areas of knowledge and specific image processing methodology. The areas correspond to the physics of the SAR imaging scenario, rules of thumb within the SAR scenario, the feature specific image characteristics, and rules of thumb for roads as imaged by microwave sensors. These areas of knowledge are detailed in the Figures 3.1 through 3.4.

5.1.1 SAR Imaging Physics

This section describes those phases of the symbolic model that concern the physics of the SAR imaging scenario. This area of knowledge embraces properties of the physics including the center wavelength of the radar, the range and azimuth resolution of the sensor, the pixel size of the final image product, the look angle, the squint angle, the signal-to-noise ratio of the SAR sensor, the SAR platform, for instance airborne or spaceborne, and the polarization signature if available.

These rules attempt to quantify the expected response of the features of interest given particular SAR parameters. Thus, these rules create a model of the imaging mission as it relates to the features of interest. Some rules in this area of knowledge relate to various features and situations so that rules in other areas, such as rules of thumb, can be more effective. For example, the parameters of the SAR scenario may be used to determine if resolution and pixel size are sufficient to distinguish a railroad paralleling a road, while a rule of thumb uses the outcome of the first rule and another reasoning ploy to determine which of the signatures is actually the road.

The following lists rules pertaining to SAR imaging physics by general subject to which they apply. Each description includes the statement of the meta-rule followed by the reasoning used, and references if appropriate.

<u>Terrain Effects (highlighting)</u>

If (DEM available) and
 (average distance of cued feature to a zero crossing is less than 5 pixels) then
  (terrain induced effects are confirmed) and
  (cued feature is discarded).

If (DEM not available) and
(cued feature is adjacent to a region of shadow along the down-range side, then
(terrain induced effects are confirmed) and
(cued feature is discarded).
If (feature avg. intensity is greater than a given threshold) and
(cued feature consists of a single linear signature) then
the certainty factor of (cued feature is a terrain effect) is high
Else,
the certainty factor of (cued feature is a terrain effect) is low.
If (feature is characterized by sharp edge gradient along its length) then
the certainty factor of (cued feature is terrain effect) should be increased.
If (feature is characterized by local contrast) then
the certainty factor of (cued feature is terrain effect) should be increased.
If (feature is adjacent to shadow) then
the certainty factor of (cued feature is terrain effect) should be increased.

In SAR images, the existence of significant terrain variations can lead to terrain effects including foreshortening, layover, and slope induced highlighting. Some image processing discriminators may incorrectly flag these effects as features of interest. For example, a sharp straight ridge may be oriented such that the steepest slope is parallel to the flight path. This slope is likely to become highlighted due to multiple radar returns from the slope to the same range gate. As a result, the slope edge may be misclassified as a road due to its width and statistically consistent pixel values. The knowledge base is therefore required to have rules that check for slope induced terrain effects in order to differentiate them from features of interest.

There are very few ways to determine if a cued feature is a terrain induced effect, or a true feature of interest. A terrain induced edge is often followed by a shadow in the down-range direction in the radar image. To use such a criteria, there must be a definition for a region of shadow. Not all terrain induced highlights, however, are adjacent to shadow. A more reliable method would require a registered digital elevation model of the region. Then the location of a flagged feature could be compared to the elevations in the DEM to determine if it lay along a ridge line. This could be determined by examining the proximity of the edge to the zero-crossings of the first difference of the DEM using a distance array image. The distance threshold should be based on the steepness of the slope of the ride whose zero-crossings are being examined. Thus the distance threshold can be set as the distance between the zero-crossing of the first-difference and the nearest local maximum of the first difference of the DEM. By this method, unfortunately, all features of interest lying along or near a ridge line would be dismissed as terrain induced effects.

<u>Polarimetry</u>
If (look angle $\geq 60^\circ$) then
surface cover is more tonally consistent regardless of polarization.
If (look angle $\leq 30^\circ$) then
surface cover is characterized by greater contrast in cross-polarized imagery.
If (region is characterized by predominantly odd # of reflections) then the certainty of
(the feature is ocean) or
(the feature is agricultural field) should be increased.
If (region is characterized by predominantly even # of reflections) then the certainty of
(the feature is urban area, or

(the feature is other corner reflector) should be increased.

If (region is characterized by predominantly diffuse reflections) then the certainty of (the feature is a park or open grassy region such as that surrounding a freeway interchange) should be increased.

If (region is characterized by a mixture of reflections) the certainty factor of (the feature is a tree covered region) should be increased. [Van Zyl, 89]

<u>Antenna Effects (striping)</u>

If (summed profiles of pixel brightnesses in image along direction of flight path are characterized by a sinusoidal variation in the range direction with constant period, or monotonacally increasing period where the first derivative of frequency with distance $\partial f/\partial s$ is linear) then (antenna effects are confirmed).

If (antenna effects) then (linear features of interest parallel to the flight path cannot be characterized by local statistical deviation).

SAR images are often characterized by azimuthal striping in the image. This effect is caused by the side lobes in the antenna pattern of the sensor. In the context of the rule-based system, it is required to differentiate these antenna effects from features of interest. Antenna effects are characterized by a low frequency sinusoidal variation in brightness in the range direction leading to the appearance of striping in the along track direction. It is possible that some of the image processing classification algorithms could misidentify the striping effects as features of interest. The knowledge base is therefore required to have a rule that checks for the characterization of antenna striping and rules out the corresponding feature.

<u>Roughness</u>

If (road type is asphalt) then ($\Delta h \approx 0.0075$ m).

If (road type is concrete) then ($\Delta h \approx 0.0015$ m).

If ($\Delta h \geq \lambda/8\cos\theta$) then (surface is rough).

If ($\Delta h \leq \lambda/8\cos\theta$) then (surface is smooth).

Concrete roads are usually characterized by large very smooth slabs consisting of a periodic linear or swirling height variations on the order of 0.15 cm in depth and 0.3 cm in period. The slabs are punctuated by expansion joints approximately every 5 meters on the order of 1-2 cm in depth. Asphalt roads are more consistent overall in smoothness with random variations on the order of 0.5 to 1 cm in depth. The values for Δh were derived from averages of these observed variations. Values of Δh for other terrain surfaces must be derived.

The roughness criteria above is based on the Rayleigh criteria where θ is the radar incidence angle on the feature. If the feature is assumed to be level and on the surface, θ can be assumed equivalent to the radar look angle.

5.1.2 SAR Rules of Thumb

This section comprises rules of thumb pertaining to SAR imaging. There is no comprehensive list of properties involved with this symbolic model, rather this area of knowledge is intended to provide a niche for general heuristics.

The organization is similar to section 5.1, containing rules arranged by subject. Meta-rules are followed by an explanation of the reasoning behind them.

Flightpath Orientation

If (terrain is urban) then
(orientation is north-south) or
(orientation is east-west).
If (flight orientation is ϕ (clockwise from due north)) and
(terrain is urban) then
(road orientation (in image) is $(\pi/2 - \phi)$ or $(\pi - \phi)$).

The angles presented in the above equation represent a local rotation transformation. This rule of thumb applies to modern urban areas in which streets and roads often follow a north-south-east-west grid system. Given the orientation of the flight path with respect to the compass directions, the orientation of the street grid within the image can be determined.

Image Artifacts

If (feature is characterized by linear signature) and
(periodic highlights) then
(feature is likely a moving target in the image).

Because of the coherent combination of signals in the synthetic aperture, a moving target will create a unusual signature. If the target is moving exactly perpendicular to the flight path direction, the object will be imaged normally, but will be displaced in the image from its stationary surroundings. For example, an automobile traveling toward the sensor would be imaged as an automobile, but at a location displaced in azimuth by an amount equal to the Doppler variation introduced by its velocity. An object moving across the field of illumination causes a similar effect except that the object may appear in multiple synthesized apertures. Thus its signature would be smeared and is usually characterized by periodic highlights. It is important to be able to distinguish this sort of feature from a transmission line, for example. A transmission line is often characterized by a piecewise linear signature punctuated by towers. Often the towers and the overhead lines will be located in areas cleared of underbrush and trees.

5.1.3 Feature Specific Image Characteristics

This section comprises a discussion of the symbolic model based on characteristics of the feature of interest separate from the imaging scenario. These characteristics can include roughness, terrain, type of road, width, and curvature. The goal being to create a model of the expected signature given a description of feature of interest, in contrast to section 5.1.1 in which the expected signature was modeled given the SAR parameters. For example, roads between agricultural boundaries are usually linear and are often gravel or dirt in consistency. A feature specific rule embracing this information could be:

if (signature is linear) and

(surface consistency is gravel or dirt) and
(linear signature divides regions of differing texture) then
the certainty of (agricultural boundary) is increased.

Clearly, this rule cannot stand alone and must depend on rules from other areas such as imaging physics or rules of thumb to differentiate surface consistency, and textural differences. This section also contains rules that use image characteristics include pixel intensity, local statistics, and gradient attributes.

As above, the following rules are organized by general subject and include a description of the background reasoning.

Roads (general)

If (terrain is desert) or
(terrain is urban) then
(roads will be characterized by long linear segments).

In flat open terrain roads and byways are usually characterized by linear segments. The exceptions to this rule occur when a road is added to an urban landscape after other features are already in place. Also, residential areas often exhibit nonlinear road segments. Ideally, a system for automated change detection in a military scenario would not be interested in such residential streets. However, in a similar system for automated GIS updating, such changes would be of foremost importance.

If (terrain is mountainous) then
(roads are characterized by curvilinear segments).

In mountainous or hilly terrain, roads often follow close to the contours of the terrain, crossing contour lines slowly to achieve a moderate inclination of the roadway. In this case, the road segments are likely to be short linear sections, coupled by more highly curved segments. Also, mountain roads tend to follow drainage lines. In this case, few or no linear segments are present in the road image primitives, rather a series of coupled curvilinear "S" segments constitute the features.

If (terrain is agricultural) then
(roads are characterized by linear segments).

In agricultural areas, roads tend to follow the paths between the fields. Due to the nature of property lines, and of planting and harvesting, these boundaries tend to be linear segments.

If (terrain is hilly) and
(road type is asphalt or concrete) then
(road will be characterized by cuts and fills).

If (terrain is characterized by cuts and fills) then
(radar signature will be characterized by alternating
corner reflections and specular reflections).

If (linear segment is constant width) and
(statistically similar pixel values) and
(terrain effects is false) then

(certainty factor of road is increased).

Once a vectorized linear segment is identified in a SAR image, and the boundaries of the region likely to contain the road are defined, this rule provides some analysis as to the certainty factor of the cued feature being a road. If the region surrounding the cued linear segment exhibits small statistical variance along a constant width section, it is likely that the segment represents a section of a road. This theoretical assumption comes from the fact that roads are usually constant width, and that any linear road segment at approximately the same distance from the imaging sensor is going to provide comparable backscattered power to the radar. Thus along the length of the cued segment, the pixel value should not statistically vary much with the region representing the road. This rule could serve to define the width of the road given that the cued segment is a road, or to determine if the cued segment is a road given that the width is known.

If (surface cover is dense) and
(canopy is trees) then
(roads will be characterized by tree lines).

In densely forested regions, roads must be maintained with regions of clear cut, such that the total cleared wodth is broader than the actual road. In SAR imagery, the clear cut region is more likely to be the dominant mechanism in the backscattered energy. This is due to the corner reflector-like effect of the flat even road next to the dense "wall" of tree trunks, branches and canopy. The location and orientation of the road with respect to the sensor as well as the width of the clear cut will also be major factors in determining the visibility of the cut. The corner reflector-like mechanism may be discernible using the unsupervised classification scheme proposed by Van Zyl. [Van Zyl, 89]

If (indicated region is characterized by a bimodal histogram) and
(the statistically darker region is characterized by a closed contour) or
(the darker region is characterized by a contour that intersects the
down-range edge of the region) then
(shadowed region is confirmed) and (vectorized border of the region can be defined).

In a SAR image a region of shadow is characterized locally by a bimodal histogram. [Curlander et al, 89] Therefore to delineate the edges of a shadowed region in a SAR image, only a rectangular region containing the suspected shadow must be indicated. The histogram of the pixel brightnesses will indicate an intensity threshold. The resulting binary image may then be vectorized to produce outlines of shadowed regions within the rectangular window.

If (terrain is desert) and
((road type is gravel) or (road type is dirt)) then
(road pixels will not be characterized by variation in relative dielectric constant).

In unvegetated or sparsely vegetated terrain, dirt or gravel roads are often constructed from local materials, and are compressed and graded to form a passable surface for vehicles. Under these conditions, the radar backscatter from the road surface compared to that of the surrounding region will not be detectable using local contrast image processing operators.

If (road type is dirt) and
(water content is high) and

(terrain is agricultural) then
(road pixels will not be characterized by local contrast).

In agricultural regions characterized by large amounts of rainfall, dirt or gravel roads between fields are likely to become damp, or muddy. Under these conditions, the radar backscatter from the road is not likely to be easily discernible from that of an unplanted field. Therefore, local contrast operators should not be used as image processing discriminators. Similarly the following rules indicate the use of local contrast operators.

If (road type is concrete) and
(terrain is agricultural) then
(road pixels may be characterized by local contrast).
If (road type is asphalt) and
(terrain is agricultural) then
(road pixels may be characterized by local contrast).

These two rules are partial converses of the two immediately above them.

The following rules apply to features that are likely to have similar signatures to roads in SAR imagery. These features may be confused with roads if only SAR physics or SAR rules of thumb are used to distinguish roads. The following rules are intended to eliminate certain features by increasing the probability that a cued object is a feature other than a road. One advantage of this approach is that the number of conflicts and false identifications is reduced.

The format of each of the rules below is to indicate increased or decreased certainty factor concerning the cued feature. Each section contains a set of essential requirements that if met can determine an initial certainty factor for the cued feature. The essential rule is numbered 0, and must be executed initially. Several rules follow each initial determination. These rules are intended to be checked after the initial rule is instantiated, and can be used to drive the likelihood to a TRUE or FALSE determination if necessary. For example, the first subject below is agricultural field boundaries. The initial rule indicates that if the cued feature is linear and characterized by differing textures on either side, then the initial certainty factor that the feature is a crop boundary should be high. If this is not the case, the certainty factor that the feature is a crop boundary is low. The following three rules can be used as a more exact indication of the status of the feature, if the initial one was not enough. If one of the secondary rules is instantiated, however all should be so that the resulting certainty factor is not biased.

Agricultural Field Boundaries

0. if (feature consists of a single linear signature) and
(feature is characterized by different textures on either side) and
the certainty factor of (cued feature is crop boundary) is high
otherwise
the certainty factor of (cued feature is crop boundary) is low.

1. if (feature is characterized by sharp edge gradient along its length) then
the certainty factor of (cued feature is crop boundary) should be increased.
2. if (feature region is characterized by bimodal histogram) then

the certainty factor of (cued feature is crop boundary) should be decreased.
3. if (feature region is characterized by odd # bounces) then
the certainty factor of (cued feature is crop boundary) should be increased.

A sharp edge gradient indicates a change in radar reflectance characteristics. Such a change in the context of having instantiated the initial rule would generally indicate the existence of some feature. When different textures are present, the gradient may be due to the change in textures, when a tonally consistent background exists, a sharp gradient may indicate a feature that is otherwise undetectable.

The bimodal histogram is indicative of a shadowed region [Curlander et al, 89], but a shadowed region is not consistent with agricultural field boundaries. The presence of a bimodal histogram would suggest the cued feature is a terrain effect, or other feature that would cast a shadow.

Scattering consistent with an odd number of bounces can be associated with agricultural areas. Consider a field of corn for example. The radar reflections can be modeled by several scattering mechanisms. The incident radiation can be directly reflected from the upturned leaves or directly from the ground. This accounts for strong single bounce reflections. The incident radiation can also be reflected by the ground onto the stalks and back to the sensor, or directly from the stalks. The latter two mechanisms are likely to be attenuated as the radiation must be propagated through the top layer of leaves to reach the imaged target. Direct reflections from the ground beneath the plants is also attenuated, but since crops are usually planted in orderly rows, approximately 50% of the ground surface is unobstructed. Thus single-bounce mechanisms dominate.

<u>Transmission Lines</u>
0. if (feature consists of a single linear signature) and
(feature signature is punctuated by periodic highlights) then
the certainty factor of (cued features is transmission line) is high,
otherwise,
the certainty factor of (cued features is transmission line) is low.

1. if (feature is characterized by sharp edge gradient along its length) then
the certainty factor of (cued feature is transmission line) should be increased.
2. if (feature is characterized by shadow adjacent to it along its length) then
the certainty factor of (cued feature is transmission line) should be increased.
3. if (feature is characterized by motion with respect to DEM) then
the certainty factor of (cued feature is transmission line) should be decreased.

Again, sharp edge gradients and shadows are used to indicate characteristics associated with transmission lines. The third characteristic, motion, can be used to determine if the imaged feature is a river, or possibly a railroad.

<u>Unimproved Roads</u>
0. If (feature is characterized by a single linear signature) and
(width of feature is 2 or fewer integer constant multiples of the width of a lane given the current resolution) then
the certainty factor of (cued feature is an unimproved road) is high.
otherwise,
the certainty factor of (cued feature is an unimproved road) is low.

1. If (certainty factor of cued feature is a crop boundary is high) then
   the certainty factor of (cued feature is an unimproved road) should be increased.
2. If (surface roughness along linear cued feature is classified as rough) then
   the certainty factor of (cued feature is an unimproved road) should be increased.
3. If (region of cued feature is characterized by a bimodal histogram) then
   the certainty factor of (cued feature is an unimproved road) should be decreased.
4. If (region of the cued feature is characterized by a high density of corner reflectors) then
   the certainty factor of (cued feature is an unimproved road) should be decreased.

This collection of rules takes advantage of previously determined results. Namely if a linear feature is identified to lie along the boundary between agricultural fields, then there is an increased certainty factor that the feature is an unimproved, or unpaved road. The occurrence of paved roads along agricultural field boundaries is much more rare than unpaved roads. Also, the roughness of the surface is a major contributor to the classification of the type of road once the feature is identified as a road. Again the occurrence of shadowed regions is correlated with the other types of features, and if present should reduce the certainty factor of a road. A high density of corner reflectors may imply the existence a numerous man-made structures. Depending on the size and number of structures, the certainty factor of an unimproved road accessing the region may be reduced. The occurrence of a large number of corner reflectors may also indicate a natural response such as from a rock strewn field. As such, the decrease applied to the certainty factor for roads due to density of corner reflectors should be small.

<u>Divided Highways</u>

0. If (feature is characterized by two parallel linear signatures whose separation
   varies by less than some variance threshold given current resolution) and
   (width of each linear signature is equal to an integer multiple of the width
   of a lane given current resolution) then
   the certainty factor of (cued feature is a divided highway) should be increased,
   otherwise,
   the certainty factor of (cued feature is a divided highway) should be decreased.

1. If (both linear signatures of feature are characterized by low intensity variation
   along their length) then
   the certainty factor of (cued feature is a divided highway) should be increased.
2. If (feature is characterized by local regional contrast) then
   the certainty factor of (cued feature is a divided highway) should be increased.
3. If (surface roughness along linear cued feature is classified as smooth) then
   the certainty factor of (cued feature is a divided highway) should be increased.
4. If (surface analysis indicates cued feature has low water content) then
   the certainty factor of (cued feature is a divided highway) should be increased.

The first rule above concerns the expectation that the scattered radiation from a road surface should not change appreciably with cross range or cross-azimuth variation. That is, the scattered power from the road surface is dependent on the surface characteristics of the road. Assuming the surface characteristics do not change appreciably, the scattered power should not change appreciably. Thus if a linear signature is present in the image, and the scattered power from it is essentially constant across the image, then the certainty that it is a road should be increased. More importantly, if this is observed, the certainty should be increased a lot in comparison to the other rules for divided highways.

For example, the second rule concerns the presence of local regional contrast. This would occur if the road surface produced sufficiently contrasting response compared to its immediate surroundings, and the road itself was narrow (one or two pixels wide) in the image. Clearly, these are very tight requirements and may not occur in high resolution imagery, or in particular environmental conditions. Also, these conditions may not indicate a road surface in all cases. Therefore, the resulting modification of the certainty factor should be smaller than for the first rule.

The third rule is based on the fact that divided highways are often constructed of asphalt or concrete or some other very smooth material. Because the surface is host to high speed vehicle travel, smoothness is a requirement. We can use this fact in reverse, namely that if a linear feature is found to be smooth, then the certainty that the feature is a highway should be increased.

Urban Grids

0. If (region contains multiple linear features) and
   (linear features are parallel and occurring at equal intervals) then
      the certainty factor of (cued feature region is urban grid) is high,
   otherwise,
      the certainty factor of (cued feature region is an urban grid) is low.

1. If (width of each linear signature is greater than or equal to twice the width of a lane given current resolution) then
   the certainty of (cued feature region contains an urban grid) should be increased.
2. If (Mueller analysis indicates feature region contains signatures characterized by even number of bounces) then
   the certainty factor of (cued feature is an urban grid) should be increased.
3. If (cued feature is characterized by local regional contrast) then
   the certainty factor of (cued feature is an urban grid) should be increased.
4. If (region growing analysis indicates dark linear signatures within bright background) then
   the certainty factor of (cued feature is an urban grid) should be increased.

Rule 2 above is derived from the scattering properties associated with urban regions of San Francisco imaged by a 24cm multi-polarization radar. The city streets were surrounded by buildings providing a large number of corner reflectors. As a result, the scattering was dominated by an even number of reflections as the incident radiation bounced off the streets and the sides of the buildings to return to the sensor. A few exceptions to these observations occurred on streets that were oriented nearly perpendicular to the incident radiation (45°). Regions of the city with such angled streets were characterized dominantly by an odd number of reflections [Van Zyl, 89]. The third and fourth rules also takes advantage of the mass of corner reflectors often observed in urban landscapes. Wide streets passing through the clutter caused by corner reflections are likely to stand out from their background as dark linear streaks in a bright background. Only streets oriented perpendicular to the flightpath, however, will generally be visible.

5.1.4 Rules of Thumb for Roads

This section presents a symbolic model, based on road characteristics, interactions, and primitives. Specifically, this section includes heuristic knowledge of roads that does not fall into either physics, or image characteristics. For example, roads intersect with other roads. Thus if one road is found, it is logical that simple edge following algorithms could be used to cue other potential road features for evaluation by other rules in the system. Also, it is common knowledge that roads generally intersect at right angles. This is not always the case, however, the certainty factor corresponding to features that do fit such a description can be increased. An detected intersection between one feature known to be a road and an unknown feature can increase road certainty factor of that feature.

An orthogonal intersection is one example of a geometric primitive of roads. Other geometric primitives include various types of freeway limited access interchanges including the cloverleaf, the trumpet, and the diamond. Detection and classification of these primitives can also lead to the location of roads. In addition to geometric primitives, there are also textural primitives and competitive primitives. Textural primitives include more general characteristics including local contrast, radiometric reversal, speckle noise, and terrain highlighting. Competitive primitives are those that compete with road primitives and include features like river and road intersections, river confluences, tree boundaries, crop boundaries, transmission lines, etc. In addition to these there are other difficult special cases for which rules should be created as required. Special cases may include perpendicular river confluences, a dam with a road over it, a river and a road in parallel, a river delta. Some examples of primitives and special cases are shown in Figures 5.1 - 5.5.

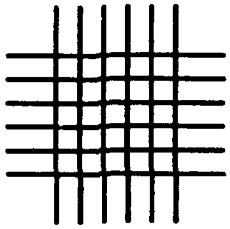
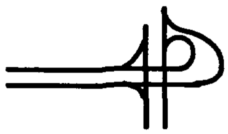
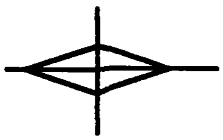
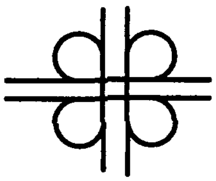
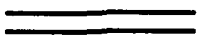
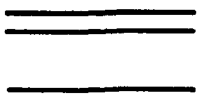
Definition	Graphic Example	Distinctive Characteristics
urban grid		all linear segments, constant spatial frequency, high density of corner reflectors dominates, roads dark in varying bright clutter
trumpet interchange		linear and curvilinear segments interchanges dominated by open area within clutter
limited access interchange		all linear segments surrounded by grassy or open area
cloverleaf		linear and circular segments surrounded by grassy or open area
secondary road		linear segment
secondary road		curvilinear segment
divided highway		parallel linear segments, constant spacing
divided highway		parallel curvilinear segments, constant spacing
divided highway		parallel curvilinear segments constant spacing
divided highway with service road		2 parallel linear segments constant spacing, 1 parallel linear segment at different spacing

Figure 5.1 Geometric Road Primitives

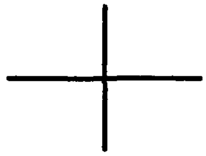
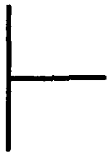
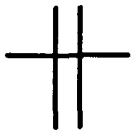
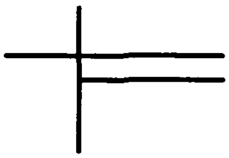
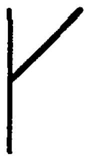
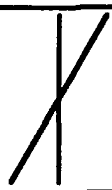
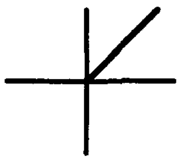
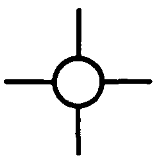
Definition	Graphic Example	Distinctive Characteristics
secondary road intersection		edge gradient, local contrast may be surrounded by clutter
secondary road intersection		linear, edge gradient, 2 or more lanes
secondary road intersection		linear, edge gradient, 2 or more lanes
secondary road intersection		linear, edge gradient, 2 or more lanes
secondary road intersection		linear, edge gradient, 1 or more lanes
secondary road intersection		linear, edge gradient, 1 or more lanes
secondary road intersection		linear, edge gradient, 2 or more lanes
traffic circle		curvilinear, edge gradient, 1 or more lanes

Figure 5.2 Geometric Intersection Primitives.

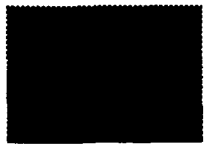
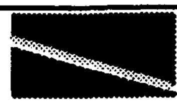
Definition	Graphic Example	Distinctive Characteristics
Tree Boundary		no discernible narrow width feature, small gradient variation
Tree Boundary		no discernible narrow width feature, diffuse polarimetric scatter
Crop boundary		linearity, local contrast edge gradient odd # of bounces
River		non parallel sides, motion of water may cause displacement from drain line in DEM
River confluence		non parallel sides, motion of water non-perpendicular intersection
Transmission Lines		discontinuous gray scale changing gradient
Canals		water appears displaced from ditch or concrete canal
Causeways		local contrast, corner reflectivity shadow radiometric reversal of water
Aqueducts		water appears displaced structure

Figure 5.3 Primitives that are competitive with roads in SAR imagery.

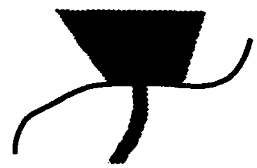
Definition	Graphic Example
Perpendicular River confluence	
Road and River	
Dam and Road	
Crops with road through	
swampy regions with many interconnected rivers	

Figure 5.4 Some cases of difficult detection cases for roads in SAR imagery.

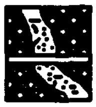
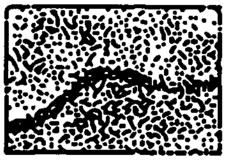
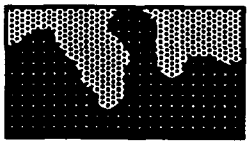
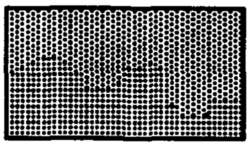
Definition	Graphic Example	Distinctive Characteristics
radiometric reversal of water		motion of water
water/land interface		motion of water
SAR shadows		locally bimodal histogram
Terrain Effects		Sharp bright edges, adjacent to shadow,close to peak in DEM
Speckle Noise		locally Rayleigh distributed histogram
Clear cuts		dense tree canopy, large narrow gradient change, # of bounces, corner reflect.
large gradient change (eg shoreline)		large gradient
small gradient change (eg tree line, crop boundary)		small gradient

Figure 5.5 Textural Primitives for SAR imagery.

This section is organized similarly to 5.1.1 - 5.1.3.

If (two features are linear) and
(they are observed to intersect only once) and
(the intersection occurs at an angle of 90°) then
the certainty factor for roads should be increased.

If (feature is characterized by a linear signature) and
(the width of the feature is an approximate multiple of the size of a lane) then
the certainty factor for roads should be increased.

If (region is urban) and
(an open area is identified to exist at the intersection of two linear segments) then
the certainty that the segments are roads should be increased, and
the certainty that the open region is a limited access interchange should be increased.

If (certainty of a limited access interchange is high) and
(identified open region is square (at any orientation)) then
the certainty that the interchange is a cloverleaf type should be increased.

If (an intersection of two divided highways is known to exist) and
(one of the highways terminates at the intersection) then
the certainty that the interchange is a trumpet type should be increased.

If (feature is characterized by linear signature) and
(the feature is exceedingly long) and
(the feature crosses a variety of terrains with curving) then
the certainty that the feature is a road should be decreased, and
the certainty that the feature is a fault line, or film scratch should be increased.

If (size of resolution cell is less than half the width of a single lane road) then
the resolution is said to be high.
If (size of resolution cell is greater than twice the width of a single lane road) then
the resolution is said to be low.

If (resolution of the image is sufficiently low) and
(feature is characterized by a sharp edge gradient) and
(feature has no discernible width) and
(variation in feature curvature is very high (ie ragged)) then
the certainty that the feature is a road should be decreased, and
the certainty that the feature is a vegetation type boundary should be increased,
Otherwise,
if (resolution is sufficiently low) and
(feature is characterized by sharp edge gradient) and
(feature has no discernible width) and
(feature is highly linear, or curvature is constant) then
the certainty that feature is a road should not be decreased.

If (a long narrow feature is identified in a tonally consistent background) and
(feature itself is tonally consistent) then
the certainty that feature is a road should be increased.

If (a long narrow feature is identified in a tonally consistent background) and
(feature itself is punctuated with specular reflections) then
the feature may be a pipeline, railroad, or transmission line.

If (terrain is not flat) and
(feature is characterized by linear signature) and
(signature crosses contour lines at a rate exceeding 25% either ascending or descending
(25 % grade = 25m increase per 100m distance))then
the certainty that feature is a road should be decreased.

5.2 Architecture of Prototype

The prototype knowledge based system did not incorporate all the rules described in Section 5.1. Six of the 20 features given in Table 3.2.1 were selected for classification in the prototype system. This resulted in a twelve layer network of 120 rules.

Each feature gave rise to two layers in the network. One layer contained the actual rules that spawned image processing and updated certainty factors. The other layer was required in order for Nexpert Object 1.1 to randomly access the rules. For a knowledge based system to work correctly, there should be no preferred or implied order in which the rules should be executed. The secondary layer for each feature classification was necessary to insure this.

The design of the prototype closely followed the rule structure derived from decision trees given in Figure 3.6. This architecture is an implementation of a frame based approach using rules. Each layer in the knowledge base contained rules concerning a particular feature type. Associated with each rule was two certainty factors or rule strengths. Depending on whether the rule was instantiated TRUE or FALSE, the overall certainty factor for the feature would be changed accordingly.

The rules in each layer did not constitute an all inclusive description of the feature, rather each layer included tests to classify candidate features by their characteristics. This is similar to a frame based approach wherein each frame consists of a list of the characteristics associated with a class. As rules were fired in the prototype, the certainty factors associated with each rule would reflect the amount that the candidate feature conformed to the overall class description.

Within each layer, rules were organized in an order dependent sequence. As a result, the order dependent certainty update rule described in section 3.4.2 was used.

6.0 Results

Section 6.1 describes the image data sets used for testing the system. Section 6.2 describes the results attained with each data set.

6.1 Description of Data sets

Four image data sets were used to test the rule/procedure system under varying conditions and for varying targets. The four data sets include SEASAT imagery of Phoenix, AZ, JPL AIRSAR of Raisin City, CA, Star-1[1] aircraft SAR of a mountainous region, and satellite imagery of the Beaufort Sea. The first two data sets contain known, mapped targets of interest that should be recognized by the system. The mountainous data set contains unmapped regions to test the system on unknown information, and the Beaufort Sea data set contains only ice floes, to evaluate the false alarm tendencies of the system. Table 6.1 summarizes the image data sets.

Table 6.1 Summary of Image Set Data

Image Name	Sensor	Pixel size	Size	Terrain Type	Polarimetric
Phoenix	SEASAT	25m	2048x2048	Agricultural/urban	No
Raisin City	JPL AIRSAR	18m	512x480	Agricultural	Yes
Beaufort Sea	SEASAT	25m	1578x1578	Ocean	No
Star-1	Intera Star-1	6m	512x512	Mountainous	No

6.1.1 Phoenix

The Phoenix data set consists of one 2048 x 2048 pixel geocoded image from rev. 523 of the SEASAT mission. The image contains regions of downtown and suburban Phoenix as well as surrounding areas of agricultural fields, mountains, and desert. The SEASAT sensor created imagery with approximately 30m resolution in both azimuth and range. The Phoenix images were resampled to 12.5m pixel size. The resulting image covered a 75km x 75km region. A subwindow of size 2048 x 2048 pixels was extracted for use at near full resolution. Various targets of interest are plainly visible in the imagery making it a good candidate for testing. In particular, cloverleaf and trumpet interchanges are apparent, as are urban road grids. Suburban nonlinear roads are lost in the clutter, but rural roads are clearly visible both between crop field and running through them at angles not aligned with the compass directions. An overview of the Phoenix data set is shown in Figure 6.1.

6.1.2 Raisin City

The Raisin City data set consists of a 1024 x 1024 pixel L-band multi-polarization image, in

1. Star-1 data sets are generated by Intera Technologies, Ltd, Calgary, Alberta, Canada.

slant range. The equivalent of a complex scattering matrix can be reconstructed at each pixel. The image was created by the JPL aircraft mounted SAR with a resolution of 7.495 m in range and 10.98 m in azimuth. The Raisin City data set was collected from an altitude of 12.22 km. The images contain predominantly rural targets consisting of agricultural fields and buildings. One main asphalt road cuts through the image from lower left to upper right, while secondary gravel roads branch off the main road between fields of crops and dirt. The Raisin City image is shown in Figure 6.2. Figure 6.3 illustrates an example of polarimetric returns.

6.1.3 Beaufort Sea

The Beaufort Sea data set consists of one 1536 x 1536 pixel geocoded image of ice floes collected on the SEASAT mission. The nominal pixel size is 100m after 4-look averaging. The predominant targets are regions of ice and water. This data set was included to test the system on imagery that positively did not contain any features of interest (i.e. roads, intersections and tree lines). An overview of the image is shown in Figure 6.4.

6.1.4 Star-1

The Star-1 data set consists of one 4096 x 4096 pixel image of a previously unmapped region. (Vexcel recently mapped the region using stereo radar with GPS.) The image was collected by an aircraft mounted X-band SAR at an altitude of roughly 10km with a nominal resolution of 6m. The pixel size was 4.8m in azimuth and 5.45m in range. The image contains predominantly uninhabited terrain containing mountains and rivers. Roads are not generally visible nor distinguishable. This data set was included to test the system on unmapped terrain and check the responses against those of a trained stereo-operator. An overview of the Star-1 image is shown in Figure 6.5.

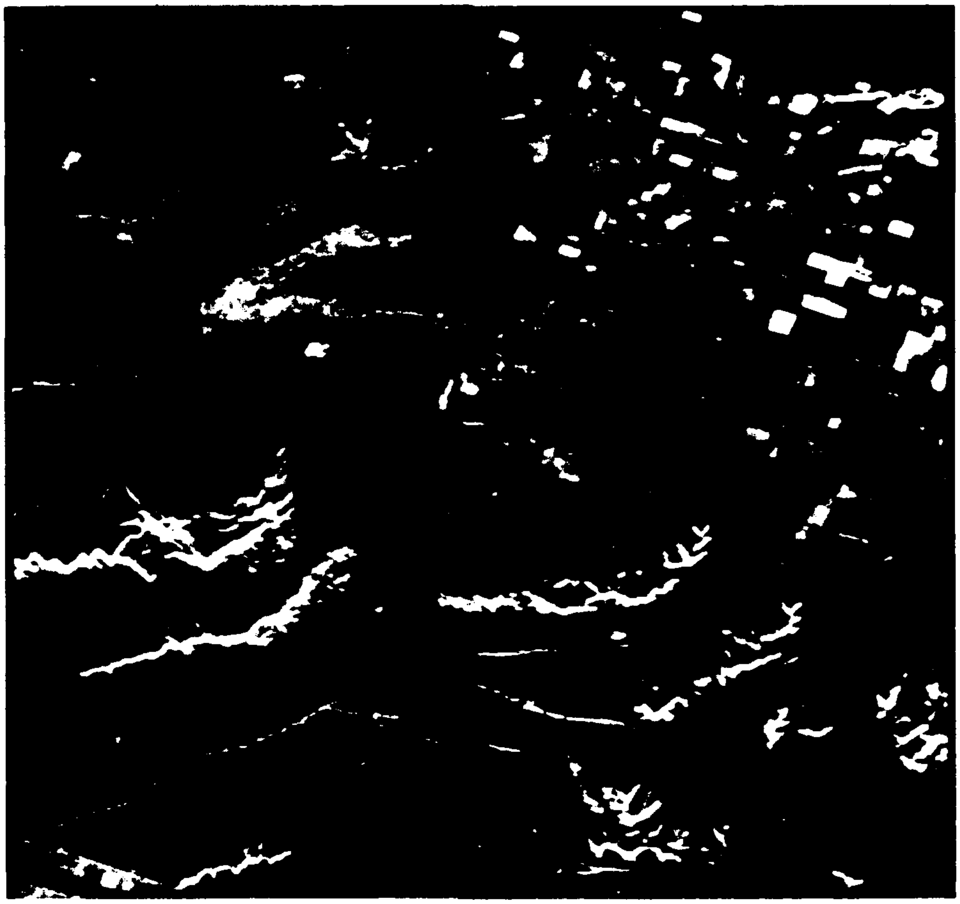


Figure 6.1 Phoenix overview image

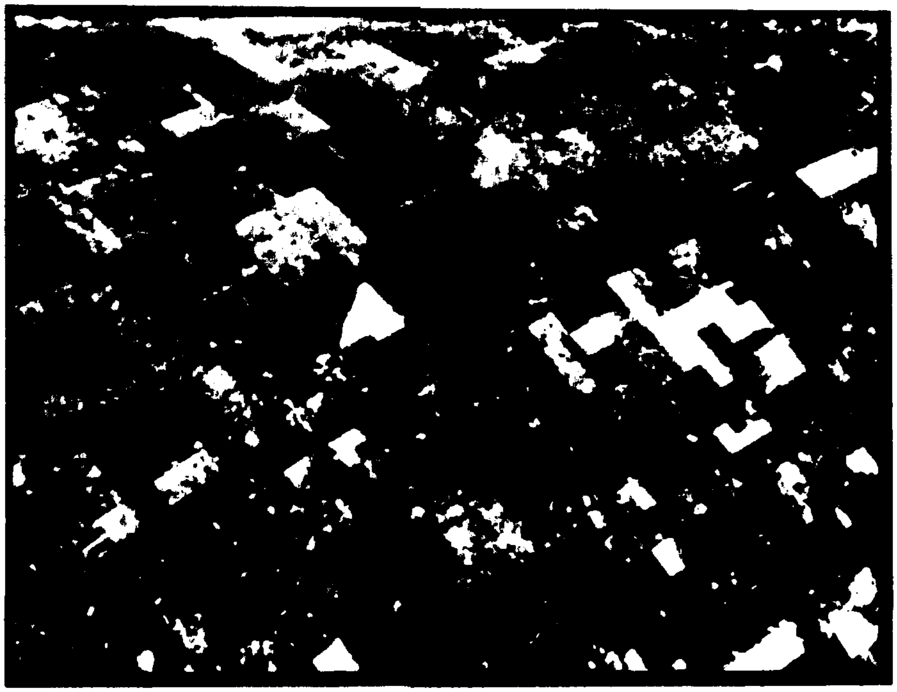


Figure 6.2 Raisin City overview image

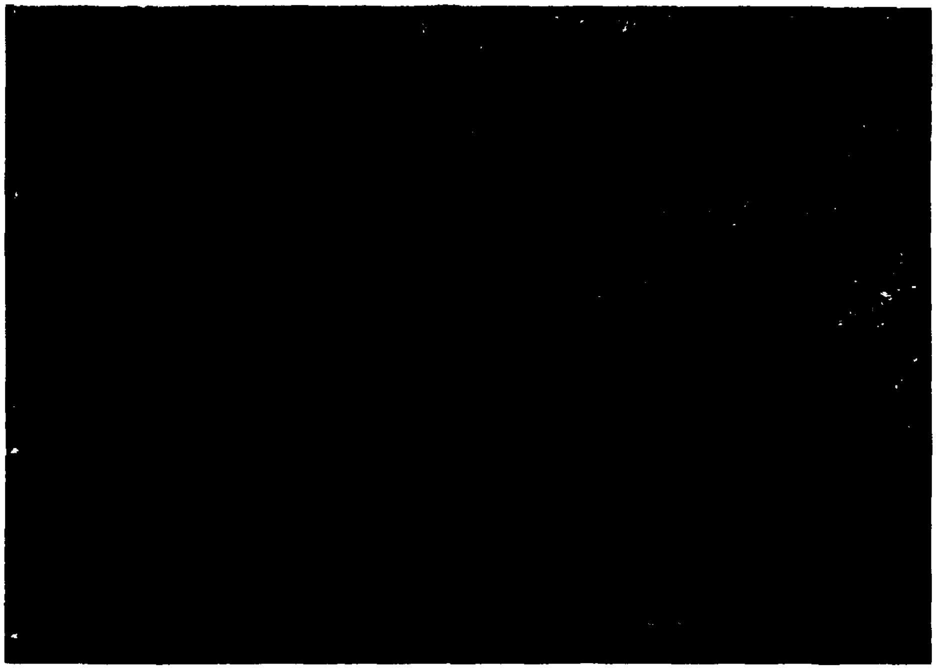


Figure 6.3 Raisin City Polarimetric returns. Red pixels indicate even number of bounces. Green pixels indicate odd number of bounces. Blue pixels indicate diffuse scattering.

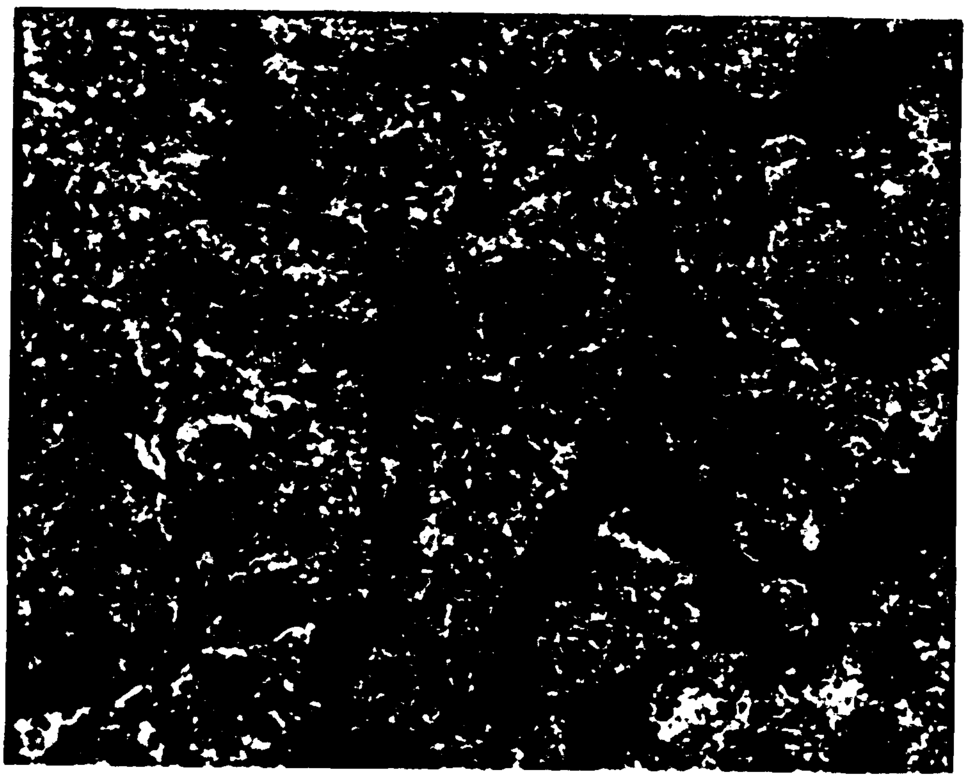


Figure 6.4 Beaufort Sea Overview image

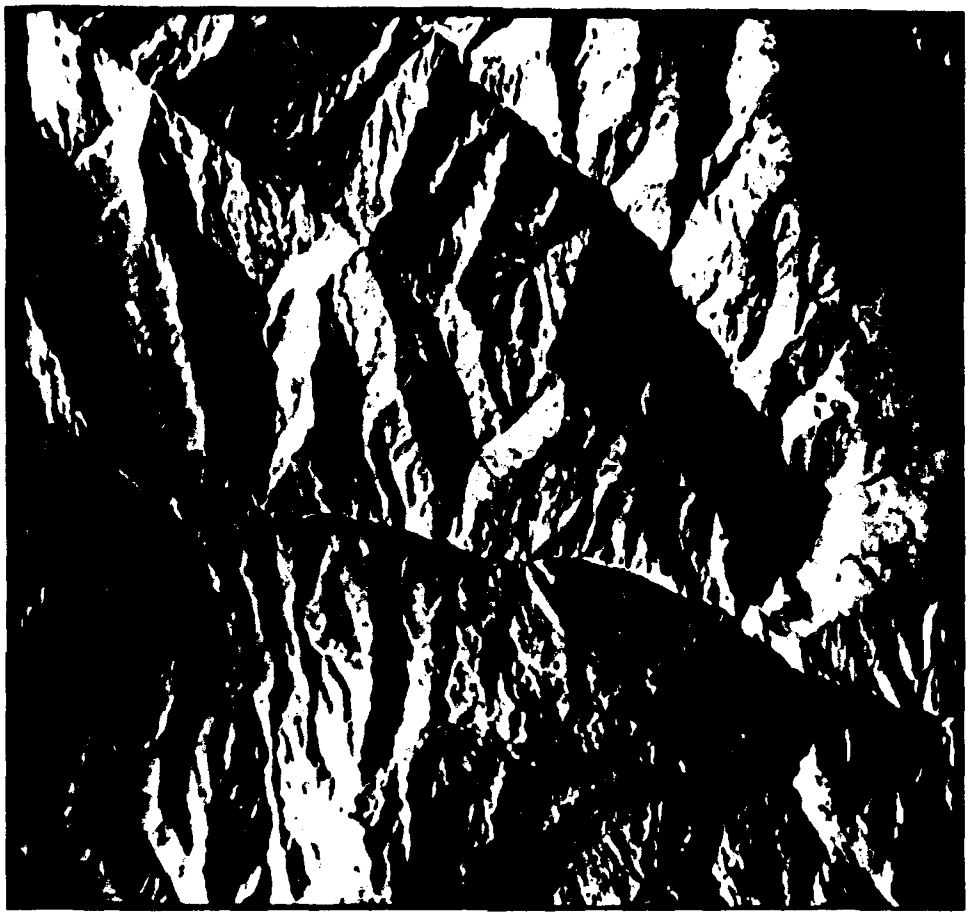


Figure 6.5 Star-1 overview image

6.2 Results of image testing

The initial image testing utilized a subset of the Phoenix SEASAT image containing chiefly rural and mountainous regions. No digital elevation model was available for use with the Phoenix data, nor was multi-polarization information. An additional test suite utilized the Raisin City imagery which contained multi-polarization signatures, the Beaufort Sea data set and the Peru data set. The Phoenix results are described in section 6.2.2 including a partial trace of the operation of the rule base. Additional results are described in the section 6.2.3.

6.2.1 Preprocessing

To provide a list of potential features to be evaluated by the rule system, a series of image preprocessing tests were enacted. This procedure was used for all test images and found to be relatively robust in that it provided a list of features of which some were of interest, and some were competing features. The preprocessing was capable of providing only partial signatures of the features that were segmented. Thus once a feature of interest was found from the cued features, a secondary procedure was required to trace the features to their end points, and find intersecting features. At that point, the knowledge processing could recommence for the intersecting features. This is one of the duties of the control executive.

In Phase II, the initial preprocessing steps will be determined by specific SAR parameters. Signal to noise ratio, resolution, pixel size, latitude and longitude, and expected types of targets may be used to delineate preprocessing methodologies.

The preprocessing consisted of a four step process. Initially, a binary edge image was created using a new edge detection process based on the pixel by pixel difference of the original image and a smoothed version of it using Lee's sigma filter [Curlander et al, 90]. A mathematical morphology (MM) opening procedure was then preformed using two simple horizontal and vertical masks [Ansoult et al, 90]. A streak detection operation is then applied to the result from the MM processing, in which vector oriented contours derived from raster image format and stored as linked lists of coordinates [Curlander et al, 90]. The longest lists are then selected as the cued image segments to be evaluated by the knowledge base. An example of the extracted feature segments used in the Phoenix testing are shown in Figure 6.6.

6.2.2 Phoenix Results and Account of Reasoning Process[1]

The initial segmentation applied to the Phoenix SEASAT provided a list of four potential features of interest to be evaluated by the system. These features consisted of a terrain highlighting effect, a transmission line, an unimproved road lying on the boundary of two agricultural fields, and a river bed.

As the processing proceeded on the Macintosh, Nexpert instructed the interactive user as to

1. Some figures in this section were generated by the Nexpert Object Development System version 1.1 and were used here by permission of Neuron Data, Palo Alto, CA. Nexpert Object is a trademark of Neuron Data Corporation.

which image processing procedures were required. For testing purposes, the operator then executed the appropriate procedures on a VAX 11/780 and Gould FD5000 image processor. The resulting information was then fed into the knowledge base as it prompted for it.

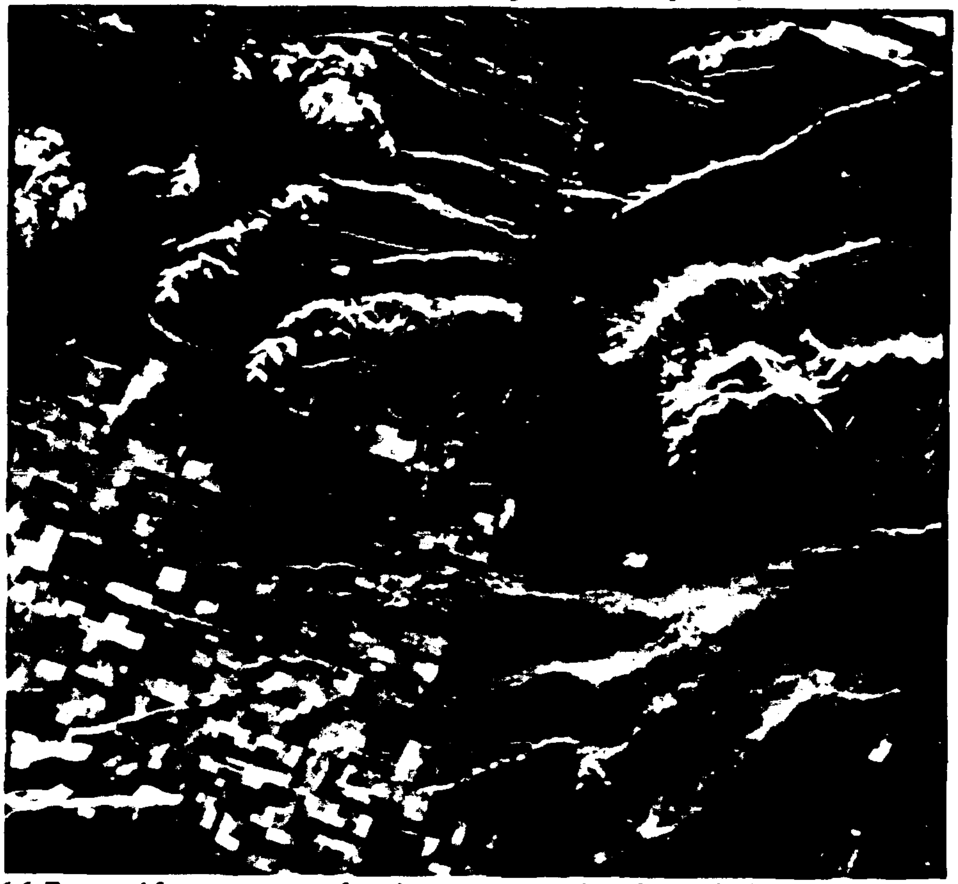


Figure 6.6 Extracted feature segments from image preprocessing of Phoenix data set.

To illustrate the knowledge processing algorithm, the following discussion will trace a few steps in the first session of the Phoenix data set. The initial feature processed in the Phoenix data set was the highlighted terrain effect. This feature was segmented due to its highly linear and bright signature. The extracted segment was 92 pixels in length.

To initialize the knowledge base, the initial certainty factors for each feature possibility were set to 0.5, the rule instantiation counter, n, was set to unity, and the hypothesis START which chains to the first rule was volunteered. In addition, a data initialization file containing the pertinent SAR information and thresholds was installed. These thresholds were derived in advance based on image parameters. Due to the arbitrary precedence of feature evaluation, the initial rules evaluate the likelihood of terrain effects. The fact that the rule chained to the terrain effects rules first was not determined by the fact that the cued feature was known to be a terrain effect. That is, there was no operator influence on the initial order of chaining of the inference engine.

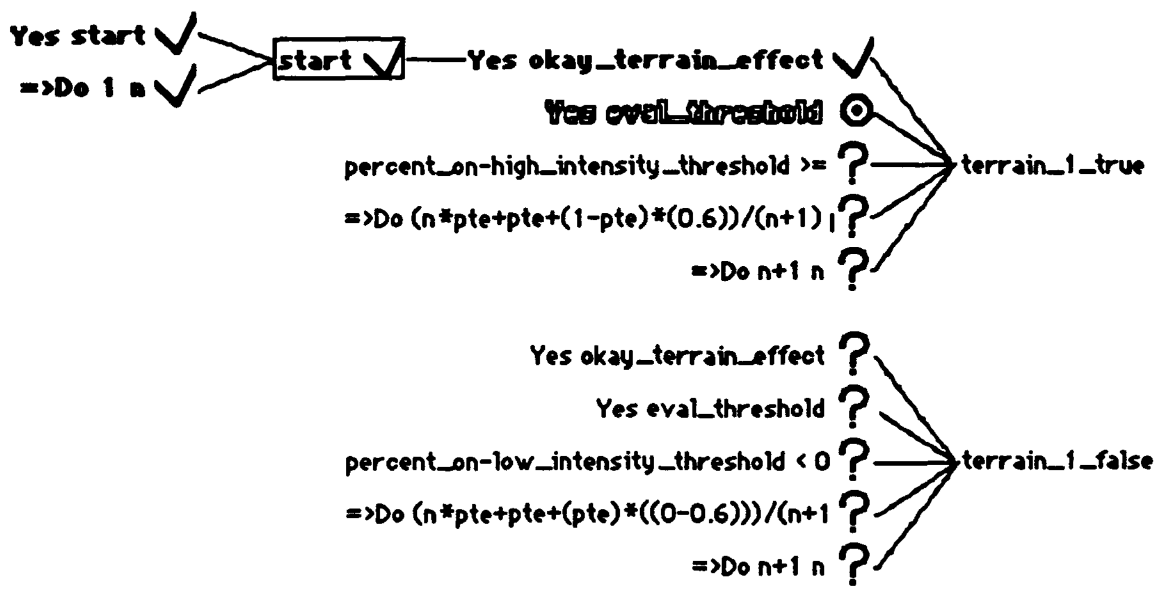


Figure 6.7 State of the rule system before **eval_threshold** is checked.

The first rule is shown in Figure 6.7, forward chained from the hypothesis START. The initial query "Yes okay_terrain_effect" is checking if the Boolean "okay_terrain_effect" has been set to true indicating that the rules concerning terrain effects may be evaluated. Setting this variable to true is a result of the START conclusion. The secondary query, "Yes eval_threshold" seeks to check the value of the Boolean "eval threshold". All Boolean values beginning with the word **eval** indicate a requirement to spawn an image processing procedure. In this case, a threshold operation is indicated. Note in Figure 6.7 the check mark indicates a TRUE instantiation, while the target indicates the rule that is currently being checked. When eval_threshold is checked, Nexpert Object will prompt the user with a question of the type "What is the value of eval_threshold, TRUE or FALSE?". At this point, the user must execute the threshold routine on the current feature using an external image processing system. When the processing is completed, the user would indicate TRUE and the rule system would look as shown in Figure 6.8.

Note, that the partial converse rule, terrain_1_false (#86) is being evaluated in parallel. This is not exactly true in Nexpert since the system is a serial processor. But when values of variables are determined, they need not be determined again for other rules.

The next query to be evaluated concerns the results of the threshold procedure. Specifically, the percentage of pixels turned on after the threshold operation is compared to the "high_intensity_threshold" which was set by the image parameters. The high_intensity_threshold is determined by the global image mean of pixel intensities. In the event that this query is evaluated to true, the following two procedures will be executed. If it is false, the procedures in the rule "terrain_1_false" will be executed. These procedures update the certainty factor corresponding to terrain effects, **pte**. As mentioned above, all certainty factors are initially set to 0.5.

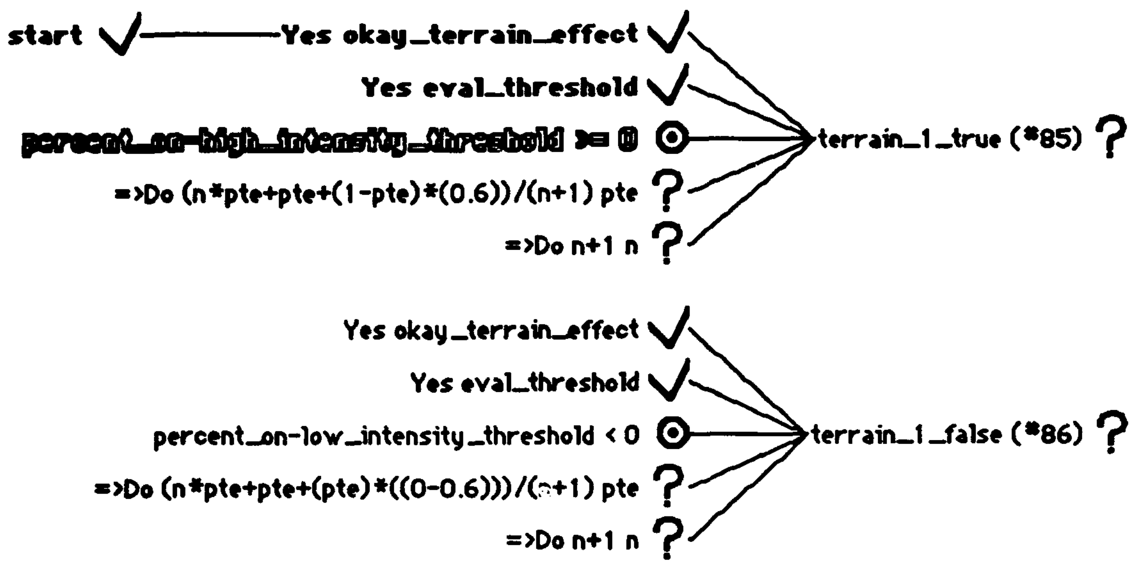


Figure 6.8 State of rule system after **eval_threshold** is set to TRUE.

As the rules are evaluated, these certainty factors are updated by the procedure mechanism shown in Figure 6.8. Note the difference between the update procedures for the true and false instantiation of the terrain_1 rule. The order dependent scoring function is being used here. A certainty factor of 0.6 has been assigned to the terrain_1 rule, resulting in a 60% increase of **pte** if terrain_1 is true, or a 60% decrease if it is false.

In this particular example, percent_on was determined to be 92.3% and the high_intensity_threshold was set at 75%. After this data was entered, the rule system looked as shown in Figure 6.9, and the value of **pte** was set to 0.65, the average of 0.5 its previous value, and 0.8 ,a 60% increase of (1 - 0.5). A complete explanation of the rule update function is given in section 3.4.2.

After the terrain_1 rule was evaluated, a Boolean variable called terrain_1 was set to true. This variable would have been set to true even if terrain_1 had been false, since it only indicates that the terrain_1 rule was evaluated. As each rule is evaluated, it provides the forward chaining to the next rule in the logical evaluation of the feature as shown in Figure 6.9.

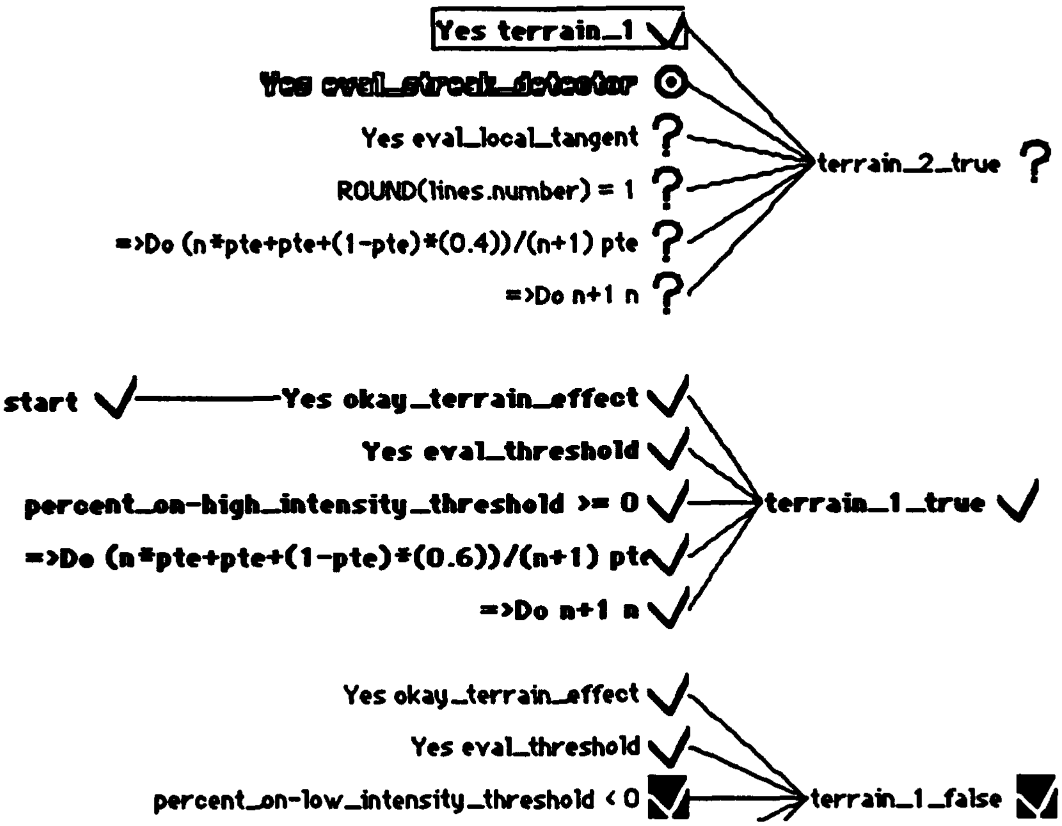


Figure 6.9. Rule system after the terrain_1 rule was evaluated.

As with the terrain_1 rule, the terrain_2 rule requires certain image processing routines to be executed, and data from those routines to be input. From Figure 6.9, it is clear that the rule terrain_2 requires the image processing routines "streak_detector" and "local_tangent" to be evaluated. After such processing, the value of "lines.number", which is determined by this processing, must be entered into the rule system. For the particular example of the terrain highlighting effect, it was found that the average number of lines was 1.14. Rounded to the nearest integer, the number of lines is unity which will cause the rule terrain_2_true to be instantiated true. At the same time, the rule terrain_2_false will be set to false, and the Boolean terrain_2 will be set to true indicating that terrain_2 rules have been evaluated, see Figure 6.10. Note the inverted check mark indicating a false conclusion to the query in the terrain_2_false rule. After terrain_2 was evaluated, the value of pte was set to 0.69.

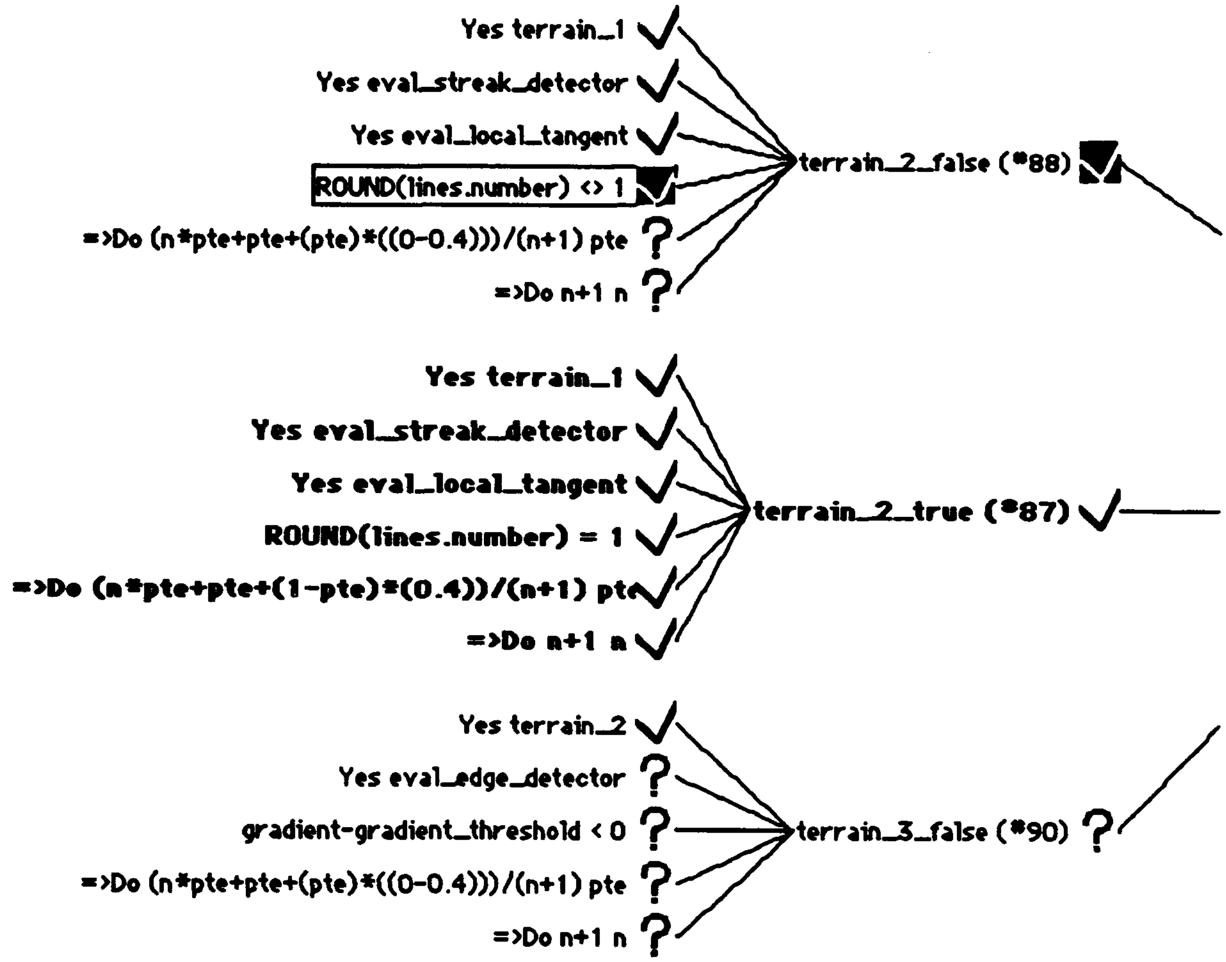


Figure 6.10. The rule system after the evaluation of the terrain_2 rules.

Evaluation of the cued feature continued as described above prompting for image processing as required, determining conclusions and updating certainty factors with the results. In addition to producing a final certainty factor for terrain effects, pte, this session also produced a certainty factor for each of the features in the rule base. That is, the rule system was capable of evaluating six features. Each session produces six certainty factors for each cued feature. Good results contain a high certainty factor for the expected feature type, and low certainty factors for the other five feature types. The resulting certainty factors for the Phoenix test are described in the next section.

6.2.3 Test Results and Discussion

The results of testing the prototype knowledge base and image processing sub-systems on the remaining features in the four image data sets described above are given in Tables 6.2 through 6.5. The certainty factors range from zero to one with 0.5 being exactly unbiased. A low certainty factor for any particular feature indicates a low chance that the candidate feature type is of that class. The prototype knowledge base was capable of distinguishing six features including: crop boundary, divided highway, terrain effect, transmission line, urban grid, and unimproved road. When presented with a feature that did fall into one of its classes, the inference engine provided a "best guess" estimate.

Table 6.2 Results of Feature Classification Testing on Phoenix SEASAT data set

Actual Feature	terrain effect	transmission line	unimproved road	river
Certainty Factors				
crop boundary	.63	.61	.60	.60
divided highway	.33	.44	.32	.32
terrain effect	**.71**	.49	.56	**.67**
transmission line	.53	**.65**	.50	.50
urban grid	.33	.47	.31	.31
unimproved road	.60	.66	**.63**	.63

Table 6.3 Results of Feature Classification Testing on Raisin City AIRSAR data set

Actual Feature	crop boundary	unimproved road	crop boundary
Certainty Factors			
crop boundary	.55	.31	.31
divided highway	.32	.32	.32
terrain effect	.56	.67	.56
transmission line	.50	.46	.46
urban grid	.31	.27	.22
unimproved road	**.66**	**.68**	**.66**

Table 6.4 Results of Feature Classification Testing on Beaufort Sea SEASAT data set

Actual Feature	edge of ice floe	edge of ice floe	edge of ice floe
Certainty Factors			
crop boundary	.5	.5	.5
divided highway	.5	.5	.5
terrain effect	**.61**	**.61**	**.61**
transmission line	.5	.5	.5
urban grid	.5	.5	.5
unimproved road	.5	.5	.5

Table 6.5 Results of Feature Classification Testing on Star-1 data set

Actual Feature	river	terrain effect	river
Certainty Factors			
crop boundary	.29	.54	.54
divided highway	.32	.32	.34
terrain effect	**.59**	**.59**	**.63**
transmission line	.49	.53	.61
urban grid	.22	.31	.33
unimproved road	**.59**	**.59**	.59

Each table shows the candidate features that were tested compared to the resulting certainty factors for each feature class produced by the expert system. The highest score is shown in bold for each candidate tested. Ideally, the boldface score should correspond to the actual feature. This is the case, for example, in the first test of the Phoenix data set. The actual feature in the image was a terrain effect. The resulting certainty factor for terrain effects was 0.71, the highest in the column. In other words, the expert system rated the candidate feature to be a terrain effect with certainty 0.71. Similarly it rated it to be a crop boundary with certainty 0.63, etc.

If the actual feature corresponded to the highest certainty factor in a test, that classification was deemed a success. If the certainty factor of all of the classified features was between 0.4 and 0.6, the classification was deemed as no bias. That is, the knowledge system was not able to gather sufficient information to make a conclusion about the data. If the highest certainty factor corresponded to a feature other than the actual feature, that was considered a failure. Since rivers, and edges of ice floes were not one of the features that the rule base could classify, those test cases were not included in the success/failure statistics. By these criteria, the prototype system was successful in 71% of the cases, and failed in 29% of the cases. The classification corresponding to ice floes always resulted in no bias solutions, while rivers were often misclassified as unimproved roads, or terrain effects.

The cases in which knowledge system failed both occurred for crop boundaries. The features were misclassified in both cases as unimproved roads. This is likely due to the rule designating unimproved roads as crop boundaries in some cases (see Appendix A). There was also a high degree of correlation between terrain effects and unimproved roads. This implies that more attributes are needed in frames dedicated to each of those features. More attributes will lead to a more exact classification between similar feature types. Addition of multi-source, or multi-spectral data sets should improve the classification problem.

Note the results for the three ice floe edge segments from the Beaufort Sea data set. Two of the three feature vectors were in excess of 100 pixels in length and all exhibited very sharp edge gradient as would be expected at the border between ice and water. The expert system, however, did not have enough data to distinguish between the features, or place them into any of the classifications that were designed in the knowledge base. This indicates that the frame based rule system is robust against input data that it is not designed to handle. The inference was fooled into thinking that these features were terrain effects, however, the resulting certainty factor is not much greater than 0.5 indicating a very weak certainty.

The results from the Star-1[1] data set indicate a rivalry between the classification of unimproved roads and terrain effects. In one case, a river was classified as as an unimproved road, and in the other as a terrain effect. Closer inspection of the image of Figure 6.5 reveals several major rivers in mountainous terrain. Comparison of the pixel values and distributions along rivers with those along nearby terrain highlighting effects illustrates the similarity between the features. The addition of a digital elevation model (DEM) would provide the necessary differentiation between these feature classes. The region shown was unmapped, thus no DEM was available. As a result, the rules pertaining to DEMs had no effect on the final results.

7.0 Conclusions

Results from testing the 120 rule prototype knowledge base indicate viability of the concept for automated target detection and identification in SAR imagery. The addition of multi-spectral, or hyper-spectral imagery would provide more information for a rule based system in Phase II, and a higher likelihood for success. Additional information provides more attributes for each feature class thus creating a more complete description for classification.

The prototype knowledge base was developed in Nexpert Object 1.1 on the Macintosh II for identification and classification of three types of roads from three competing features in SAR imagery. The prototype was tested on 16 candidate features from four SAR data sets including both satellite and airborne sensors. The JPL Raisin City data set included polarimetric data. The classification accuracy is summarized as follow:

Targets Classified Correctly:	71%
Targets Classified Incorrectly:	29%

Targets without classification frames in the prototype were classified with no bias in 50% of the cases, and incorrectly in 50% of the cases.

The primary reason for incorrect classification was ambiguity in the attribute description for each feature in the frame-based implementation of the inference engine. Also, when targets were provided to the inference engine for which no frames existed to describe them, they were incorrectly classified 50% of the time. Therefore it is important to develop frames for all possible features of interest that may be encountered in strategic situations. Further work is required to produce sufficiently distinct attribute sets for each feature of interest, possibly including hyper-spectral data attributes.

The knowledge environment should be more interactive and less automated. In the current system, potential exists for misclassification due to incomplete attribute descriptions in each frame. Even with highly detailed frame descriptions this problem will remain with fully automated systems. With a small amount of operator interaction, the incidence of incorrect classification could be radically reduced without much time lost.

More descriptive frames for feature classes of interest should be developed. Frames for unwanted classes can be sparser as long as they are sufficient to remove such features from consideration as features of interest.

Remaining problems and concerns include:

- More extensive testing and characterization of performance of the current prototype in a wider variety of SAR scenarios.
- Developing more descriptive frames for each type of feature class of interest.
- Obtaining multi-spectral and/or hyper-spectral data sets for development of classification algorithms with increased data capability.
- More fully exploiting realtionships between features, such as the interconnectivity of roads, as a technique for identification.

8.0 Recommendations for Phase II

The all-weather, day and night coverage provided by Synthetic Aperture Radar (SAR) promises an ideal source of information for target classification and tactical change detection. One problem with SAR sensors however, is a lack of consistency in imaged signatures for the same target given variations in imaging scenarios. We have shown in this Phase I effort the feasibility of using symbolic models for feature classification in a variety of SAR scenarios.

Having shown feasibility in Phase I, we propose to develop a prototype deliverable software system for Phase II that supports:

- SAR and hyperspectral image data with collateral data sets, including imaging parameters, digital feature data, DTED, etc.,
- interactive analysis on an open single workstation,
- feature classification and change analysis for one to two feature classes using symbolic modeling,
- architecture providing for symbolic modeling of additional feature classes as required.

The work plan for building such a system would include integrating / developing the following:

- an interactive custom inference engine written in Lisp,
- an integrated image processing and expert system development environment[1],
- a library of image processing and image understanding routines,
- a relational database of image and feature data,
- an interactive Unix[2], X-Window System[3] environment.

The proposed deliverable symbolic classification software will extend the capabilities of system for registration and change queuing previously contracted to Vexcel by the US Army Topographic Engineering Center. Therefore, a significant amount of the software engineering groundwork will already be completed providing more opportunity to enhance the symbolic system development.

The use of an integrated image processing and expert system development environment will alleviate the task of communication between the knowledge system and the image processing environment. Such a development environment will provide a comprehensive library of image processing and image understanding subroutines with higher level Lisp programming in

By using the Lisp programming language, we will be able to develop a custom inference engine for the task of automated classification and change detection. Rather than undertaking the modification of a rule environment suited to a different task, which proved to be difficult in Phase I, we will develop custom applications and a frame-based implementation of the classification knowledge for specific feature types.

1. A preliminary analysis shows that Amerinex Artificial Intelligence provides a commercially available system for this purpose known as KBVision.
2. Unix is a trademark of AT&T Bell Laboratories.
3. The X-Window System is a trademark of Massachusetts Institute of Technology.

The development environment will be a multi-layer tool consisting of: a set of conventional image processing tools, a library of higher level image understanding sub-systems, a relational feature database in which attributes may be defined by the user, and a Common Lisp programming environment. It will be an open system providing for user extension at all levels. In Phase II we will incorporate hyper-spectral registration and change detection algorithms, and add custom C code programs implementing new algorithms for sub-pixel registration. The Lisp programming environment will have direct access to the feature database, and can execute any procedural subroutine in the system. Thus, the control interface between the image understanding, registration and change detection, and the expert system will be invisible[1].

The Phase II knowledge processing module will have an interactive multi-color graphical user interface. The frame-based expert system will operate in concert with the user, highlighting possible targets according to the likelihood derived from the encoded domain knowledge. Features will be marked by colored outlines for clear identification by the operator. On a Unix platform, the expert assistant will carry out multiple decision tasks simultaneously providing dynamic screen updating of the information as it is processed. Since Unix is a multi-tasking environment, the operator may monitor automated processing of the expert system, or pursue other tasks. At any time, the user will be able to intervene to correct or add data to the Lisp processor.

The architecture of the knowledge module will provide for expansion so that knowledge bases for various feature types can be added to increase the functionality of the system. The Phase II effort will concentrate on development of a knowledge base for identification and classification of roads in hyper-spectral imagery. This development will draw heavily on the Phase I rules for road-type features in SAR imagery. Additional rules will be added to take advantage of hyper-spectral data sources. Provision will be made to handle additional rule systems in Lisp for other feature types.

As a tradeoff, the Phase II effort may be directed toward a broader base development of frames for classification of a wide variety of features. In that event, however, each class would be described by relatively sparse frame attribute sets. Misclassification would be more prominent, and the resulting system would essentially provide a demonstration of the possible applicability of symbolic systems in classification and change detection. It is therefore recommended that the Phase II effort concentrate on extensive symbolic and procedural development of one type of feature. The system architecture will be designed to accommodate numerous feature classification knowledge systems, but the bulk of the time will be directed toward producing a useful tool for automated classification and change detection.

9.0 References

Ansoult et al, 90 — M. Ansoult, P. Soille, J. Loodts, "Mathematical Morphology: A Tool for Automated GIS Data Acquisiion from Scanned Thematic Maps," PE&RS **56** (9), September 1, 1990.

Ballard et al, 82 — *Computer Vision*, D. Ballard, C. Brown, Prentice Hall, Inc. Englewwod Cliffs, N.J., 1982 p77.

Cheeseman, 85 — Cheeseman,P.,"In Defense of Probability," Proc. 8th IJCAI-85, Los Angeles, CA, 1985.

Colwell et al, 83	R. Colwell et al, (ed) *Manual of Remote Sensing*, Sheridan Press, 1983.
Curlander et al, 89	J. Curlander, W. Kober, "Automated SAR Image-Terrain Database Matching," IGARSS, Vancouver, BC, Canada, 1989.
Curlander et al, 90	J. Curlander, W. Kober, J. Thomas, A. Stocker, "An Automated Software System for Updating Digital Terrain Databases from All-Source Imagery," Final Report for contract DACA76-90-C-0013, Engineering Topographic Laboratory, December 1990.
Dempster, 68	Dempster, A.,"A Generalization of Bayesian Inference," J. Roy. Stat Soc., (ser. B), 1968.
Fagin et al, 88	Fagin, R., Halpern, J., "Reasoning about Knowledge and Probability," Proc. Second Conf. on Theoretical Aspects of Reasoning About Knowledge, ed M. Vardi, Morgan Kaufmann, 1988.
Gordon et al, 85	Gordon,J., Shortliffe,E.,"A Method of Managing Evidential Reasoning In a Hierarchical Evidence Space," Artif. Intell., vol-26, July 1985.
Halpern et al, 87	Halpern, J., Rabin, M., "A Logic to Reason about Likelihood", Artif. Intel., vol-32, 1987.
Heckerman, 86	Heckerman, "Probabalistic Interpretation for MYCIN's Certainty Factors," in *Uncertainty in Artificial Intelligence*, vol-2, ed L. Kanal and J. Lemmer, North Holland Press, 1986.
Hummel et al, 88	Hummel,R., Landy,M.,"A Statistical Viewpoint on the Theory of Evidence," IEEE Trans. PAMI, vol-10, #2, March 1988.
McCarthy et al, 69	McCarthy, J., P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Machine Intelligence*, ed B. Meltzer and D. Michie, vol-4, Edinburgh Eniv. Press, 1969.
McCarthy, 86	McCarthy, J. "Applications of Circumscription to Formalizing COmmon-Sense Knowledge," Artif. Intel., vol-28, 1986.
Minsky, 75	M. Minsky, "A Framework for Representing Knowledge," in *Psychology of Computer Vision*, ed P. WInston, McGraw Hill, 1975.
Pearl, 88	Pearl,J., Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, 1988.
Safranek et al, 90	Safranek,R., et al,"Evidence Accumulation Using Binary Frames of Discernment for Verification Vision," IEEE Trans. Rob. and Autom., vol-6, #4, Aug. 1990.
Shafer et al, 87	Shafer,G., Logan, R.,"Implementing Dempster's Rule for Hierarchical Evidence," Artif. Intell., vol-33, 1987.
Shortliffe, 76	Shortliffe, E., Computer-Based Medical Consultation: MYCIN, Elsevier, 1976.
Van Zyl, 89	J. Van Zyl, "Unsupervised Classification of Scattering Behavior Using radar Polarimetry Data," IEEE Trans. Geo. Rem. Sens., 27(1) Jan. 1989.
Yen, 89	Yen,J.,"GERTIS: A Dempster-Shafer Approach to Diagnosing Hierarchical Hypotheses", CACM, vol-32, #5, May 1989.
Zadeh, 86	Zadeh,L,"A simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination," AI Mag., vol-7, #2, 1986.

Appendix A List of Nexpert Encoded Rules

Rule crop_1_true (#1)
If there is evidence of okay_crop_boundary
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is less than 0
Then crop_1 is confirmed.
And (n*pcb+pcb+(1-pcb)*(0.6))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_1_false (#2)
If there is evidence of okay_crop_boundary
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is greater than or equal to 0
Then crop_1 is confirmed.
And (n*pcb+pcb+pcb*((0-0.6)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_2_true (#3)
If there is evidence of crop_1
And there is evidence of eval_local_contrast
And there is evidence of eval_DEM_drain_analysis
And there is no evidence of motion
Then crop_2 is confirmed.
And (n*pcb+pcb+(1-pcb)*(0.4))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_2_noncommittal (#4)
If there is evidence of crop_1
And there is evidence of eval_local_contrast
And there is no evidence of eval_DEM_drain_analysis
Then crop_2 is confirmed.
And n+1 is assigned to n

Rule crop_2_false (#5)
If there is evidence of crop_1
And there is evidence of eval_local_contrast
And there is evidence of eval_DEM_drain_analysis
And there is evidence of motion
Then crop_2 is confirmed.
And (n*pcb+pcb+(pcb)*((0-0.4)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_3_true (#6)
If there is evidence of crop_2
And there is evidence of eval_Mueller
And percent_odd is greater than or equal to 50
Then crop_3 is confirmed.
And (n*pcb+pcb+(1-pcb)*(0.4))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_3_noncommittal (#7)
If there is evidence of crop_2
And there is no evidence of eval_Mueller
Then crop_3 is confirmed.
And n+1 is assigned to n

Rule crop_3_false (#8)
If there is evidence of crop_2
And there is evidence of eval_Mueller
And percent_odd is less than 50
Then crop_3 is confirmed.
And (n*pcb+pcb+(pcb)*((0-0.4)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_4_true (#9)
If there is evidence of crop_3
And there is evidence of eval_edge_detector
And gradient-gradient_threshold is greater than or equal to 0
Then crop_4 is confirmed.
And (n*pcb+pcb+(pcb)*((0-0.2)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_4_false (#10)
If there is evidence of crop_3
And there is evidence of eval_edge_detector
And gradient-gradient_threshold is less than 0
Then crop_4 is confirmed.
And (n*pcb+pcb+(1-pcb)*(0.2))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_5_true (#11)
If there is evidence of crop_4
And there is evidence of eval_bimodal_histogram
And there is no evidence of bimodal_histogram
Then crop_5 is confirmed.
And n+1 is assigned to n

Rule crop_5_false (#12)
If there is evidence of crop_4
And there is evidence of eval_bimodal_histogram
And there is evidence of bimodal_histogram
Then crop_5 is confirmed.
And (n*pcb+pcb+pcb*((0-0.4)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_6_true (#13)
If there is evidence of crop_5
And there is evidence of eval_Rayleigh_histogram
And there is evidence of Rayleigh_histogram
Then crop_6 is confirmed.
And (n*pcb+pcb+(1-pcb)*(0.2))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule crop_6_false (#14)
If there is evidence of crop_5
And there is evidence of eval_Rayleigh_histogram
And there is no evidence of Rayleigh_histogram
Then crop_6 is confirmed.
And (n*pcb+pcb+(pcb)*((0-0.2)))/(n+1) is assigned to pcb
And n+1 is assigned to n

Rule divided_1_true (#15)
If there is evidence of okay_divided_highway
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is precisely equal to 2
Then divided_1 is confirmed.
And (n*pdh+pdh+(1-pdh)*0.4)/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_1_false (#16)
If there is evidence of okay_divided_highway
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is not equal to 2
Then divided_1 is confirmed.
And (n*pdh+pdh+pdh*((0-0.4)))/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_2_true (#17)
If there is evidence of divided_1
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-4*lane.size is greater than or equal to 0
Then divided_2 is confirmed.
And (n*pdh+pdh+(1-pdh)*0.6)/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_2_false (#18)
If there is evidence of divided_1
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-4*lane.size is less than 0
Then divided_2 is confirmed.
And (n*pdh+pdh+(pdh)*((0-0.6)))/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_3_true (#19)
If there is evidence of divided_2
And there is evidence of eval_local_contrast
And there is evidence of eval_DEM_drain_analysis
And there is no evidence of motion
Then divided_3 is confirmed.
And (n*pdh+pdh+(1-pdh)*0.4)/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_3_noncommittal (#20)

If there is evidence of divided_2
And there is evidence of eval_local_contrast
And there is no evidence of eval_DEM_drain_analysis
Then divided_3 is confirmed.
And n+1 is assigned to n

Rule divided_3_false (#21)
If there is evidence of divided_2
And there is evidence of eval_local_contrast
And there is evidence of eval_DEM_drain_analysis
And there is evidence of motion
Then divided_3 is confirmed.
And (n*pdh+pdh+pdh*((0-0.4)))/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_4_true (#22)
If there is evidence of divided_3
And there is evidence of eval_Mueller
And percent_water is less than or equal to 30
Then divided_4 is confirmed.
And (n*pdh+pdh+(1-pdh)*0.3)/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_4_noncommittal (#23)
If there is evidence of divided_3
And there is no evidence of eval_Mueller
Then divided_4 is confirmed.
And n+1 is assigned to n

Rule divided_4_false (#24)
If there is evidence of divided_3
And there is evidence of eval_Mueller
And percent_water is greater than 30
Then divided_4 is confirmed.
And n+1 is assigned to n

Rule divided_5_true (#25)
If there is evidence of divided_4
And there is evidence of eval_streak_detector
And there is evidence of eval_width_profiler
And there is evidence of eval_local_statistics
And intensity.sigma-low_sigma_threshold is less than or equal to 0
Then divided_5 is confirmed.
And (n*pdh+pdh+(1-pdh)*0.2)/(n+1) is assigned to pdh
And n+1 is assigned to n

Rule divided_5_false (#26)
If there is evidence of divided_4
And there is evidence of eval_streak_detector
And there is evidence of eval_width_profiler
And there is evidence of eval_local_statistics
And intensity.sigma-low_sigma_threshold is greater than 0
Then divided_5 is confirmed.
And n+1 is assigned to n

Rule divided_6_true (#27)
If there is evidence of divided_5
    And there is evidence of eval_local_contrast
    And contrast.sigma-sigma_threshold is greater than or equal to 0
Then divided_6 is confirmed.
    And (n*pdh+pdh+(1-pdh)*0.2)/(n+1) is assigned to pdh
    And n+1 is assigned to n

Rule divided_6_false (#28)
If there is evidence of divided_5
    And there is evidence of eval_local_contrast
    And contrast.sigma-sigma_threshold is less than 0
Then divided_6 is confirmed.
    And n+1 is assigned to n

Rule divided_7_true (#29)
If there is evidence of divided_6
    And there is evidence of eval_Mueller
    And there is evidence of eval_threshold
    And surface.roughness is "smooth"
Then divided_7 is confirmed.
    And (n*pdh+pdh+(1-pdh)*0.2)/(n+1) is assigned to pdh
    And n+1 is assigned to n

Rule divided_7_noncommittal (#30)
If there is evidence of divided_6
    And there is no evidence of eval_Mueller
Then divided_7 is confirmed.
    And n+1 is assigned to n

Rule divided_7_false (#31)
If there is evidence of divided_6
    And there is evidence of eval_Mueller
    And there is evidence of eval_threshold
    And surface.roughness is not "smooth"
Then divided_7 is confirmed.
    And n+1 is assigned to n

Rule grid_1_true (#32)
If there is evidence of okay_urban_grid
    And there is evidence of eval_streak_detector
    And there is evidence of eval_local_tangent
    And ROUND(lines.number) is precisely equal to 2
Then grid_1 is confirmed.
    And (n*pug+pug+(1-pug)*0.7)/(n+1) is assigned to pug
    And n+1 is assigned to n

Rule grid_1_false (#33)
If there is evidence of okay_urban_grid
    And there is evidence of eval_streak_detector
    And there is evidence of eval_local_tangent
    And ROUND(lines.number) is not equal to 2
Then grid_1 is confirmed.

And (n*pug+pug+pug*((0-0.7)))/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_2_true (#34)
If there is evidence of grid_1
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-4*lane.size is greater than or equal to 0
Then grid_2 is confirmed.
And (n*pug+pug+(1-pug)*0.3)/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_2_false (#35)
If there is evidence of grid_1
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-4*lane.size is less than 0
Then grid_2 is confirmed.
And (n*pug+pug+pug*((0-0.3)))/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_3_true (#36)
If there is evidence of grid_2
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is less than 0
Then grid_3 is confirmed.
And (n*pug+pug+(1-pug)*0.4)/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_3_false (#37)
If there is evidence of grid_2
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is greater than or equal to 0
Then grid_3 is confirmed.
And (n*pug+pug+pug*((0-0.4)))/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_4_true (#38)
If there is evidence of grid_3
And there is evidence of eval_local_contrast
And contrast.sigma-sigma_threshold is greater than or equal to 0
Then grid_4 is confirmed.
And (n*pug+pug+(1-pug)*0.2)/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_4_false (#39)
If there is evidence of grid_3
And there is evidence of eval_local_contrast
And contrast.sigma-sigma_threshold is less than 0
Then grid_4 is confirmed.
And n+1 is assigned to n

Rule grid_5_true (#40)
If there is evidence of grid_4
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And there is evidence of eval_streak_detector
And ROUND(lines.number)-1 is greater than 0
And lines.spacing_variance-line_spacing_variance_threshold is less than or equal to 0
Then grid_5 is confirmed.
And (n*pug+pug+(1-pug)*0.8)/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_5_falseB (#41)
If there is evidence of grid_4
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And there is evidence of eval_streak_detector
And ROUND(lines.number)-1 is less than or equal to 0
Then grid_5 is confirmed.
And (n*pug+pug+pug*((0-0.8)))/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_5_falseA (#42)
If there is evidence of grid_4
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And there is evidence of eval_streak_detector
And ROUND(lines.number)-1 is greater than 0
And lines.spacing_variance-line_spacing_variance_threshold is greater than 0
Then grid_5 is confirmed.
And (n*pug+pug+pug*((0-0.8)))/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_6_true (#43)
If there is evidence of grid_5
And there is evidence of eval_Mueller
And percent_even is greater than or equal to 50
Then grid_6 is confirmed.
And (n*pug+pug+(1-pug)*0.4)/(n+1) is assigned to pug
And n+1 is assigned to n

Rule grid_6_noncommittal (#44)
If there is evidence of grid_5
And there is no evidence of eval_Mueller
Then grid_6 is confirmed.
And n+1 is assigned to n

Rule grid_6_false (#45)
If there is evidence of grid_5
And there is evidence of eval_Mueller
And percent_even is less than 50
Then grid_6 is confirmed.
And n+1 is assigned to n

Rule jump_terrain_4 (#46)

If there is evidence of terrain_4
And pte is less than or equal to 0.1
Then okay_crop_boundary is confirmed.
And 1 is assigned to n

Rule jump_terrain_3 (#50)
If there is evidence of terrain_3
And pte is less than or equal to 0.1
Then okay_crop_boundary is confirmed.
And 1 is assigned to n

Rule jump_terrain_2 (#47)
If there is evidence of terrain_2
And pte is less than or equal to 0.1
Then okay_crop_boundary is confirmed.
And 1 is assigned to n

Rule jump_terrain_1 (#48)
If there is evidence of terrain_1
And pte is less than or equal to 0.1
Then okay_crop_boundary is confirmed.
And 1 is assigned to n

Rule 49
If there is evidence of terrain_1
And there is evidence of terrain_2
And there is evidence of terrain_3
And there is evidence of terrain_4
And there is evidence of terrain_5
Then okay_crop_boundary is confirmed.
And 1 is assigned to n

Rule jump_unimproved_4 (#51)
If there is evidence of unimproved_4
And pur is less than or equal to 0.1
Then okay_divided_highway is confirmed.
And 1 is assigned to n

Rule jump_unimproved_3 (#52)
If there is evidence of unimproved_3
And pur is less than or equal to 0.1
Then okay_divided_highway is confirmed.
And 1 is assigned to n

Rule jump_unimproved_2 (#53)
If there is evidence of unimproved_2
And pur is less than or equal to 0.1
Then okay_divided_highway is confirmed.
And 1 is assigned to n

Rule jump_unimproved_1 (#55)
If there is evidence of unimproved_1
And pur is less than or equal to 0.1
Then okay_divided_highway is confirmed.

And 1 is assigned to n

Rule 54
If there is evidence of unimproved_1
And there is evidence of unimproved_2
And there is evidence of unimproved_3
And there is evidence of unimproved_4
And there is evidence of unimproved_5
Then okay_divided_highway is confirmed.
And 1 is assigned to n

Rule start (#56)
If there is evidence of start
Then okay_terrain_effect is confirmed.
And 1 is assigned to n

Rule jump_crop_5 (#57)
If there is evidence of crop_5
And pcb is less than or equal to 0.1
Then okay_transmission_line is confirmed.
And 1 is assigned to n

Rule jump_crop_4 (#58)
If there is evidence of crop_4
And pcb is less than or equal to 0.1
Then okay_transmission_line is confirmed.
And 1 is assigned to n

Rule jump_crop_3 (#59)
If there is evidence of crop_3
And pcb is less than or equal to 0.1
Then okay_transmission_line is confirmed.
And 1 is assigned to n

Rule jump_crop_2 (#60)
If there is evidence of crop_2
And pcb is less than or equal to 0.1
Then okay_transmission_line is confirmed.
And 1 is assigned to n

Rule jump_crop_1 (#62)
If there is evidence of crop_1
And pcb is less than or equal to 0.1
Then okay_transmission_line is confirmed.
And 1 is assigned to n

Rule 61
If there is evidence of crop_1
And there is evidence of crop_2
And there is evidence of crop_3
And there is evidence of crop_4
And there is evidence of crop_5
And there is evidence of crop_6
Then okay_transmission_line is confirmed.

And 1 is assigned to n

Rule jump_trans_5 (#63)
If there is evidence of trans_5
And ptl is less than or equal to 0.1
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule jump_trans_4 (#64)
If there is evidence of trans_4
And ptl is less than or equal to 0.1
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule jump_trans_3 (#65)
If there is evidence of trans_3
And ptl is less than or equal to 0.1
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule jump_trans_2 (#66)
If there is evidence of trans_2
And ptl is less than or equal to 0.1
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule jump_trans_1 (#68)
If there is evidence of trans_1
And ptl is less than or equal to 0.1
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule 67
If there is evidence of trans_1
And there is evidence of trans_2
And there is evidence of trans_3
And there is evidence of trans_4
And there is evidence of trans_6
And there is evidence of trans_6
Then okay_unimproved_road is confirmed.
And 1 is assigned to n

Rule jump_divided_6 (#69)
If there is evidence of divided_6
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_divided_5 (#70)
If there is evidence of divided_5
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_divided_4 (#71)
If there is evidence of divided_4
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_divided_3 (#72)
If there is evidence of divided_3
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_divided_2 (#73)
If there is evidence of divided_2
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_divided_1 (#74)
If there is evidence of divided_1
And pdh is less than or equal to 0.1
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule 75
If there is evidence of divided_1
And there is evidence of divided_2
And there is evidence of divided_3
And there is evidence of divided_4
And there is evidence of divided_5
And there is evidence of divided_6
And there is evidence of divided_7
Then okay_urban_grid is confirmed.
And 1 is assigned to n

Rule jump_grid_6 (#76)
If there is evidence of grid_6
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule jump_grid_5 (#77)
If there is evidence of grid_5
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule jump_grid_4 (#78)
If there is evidence of grid_4
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule jump_grid_3 (#79)
If there is evidence of grid_3
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule jump_grid_2 (#80)
If there is evidence of grid_2
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule jump_grid_1 (#81)
If there is evidence of grid_1
And pug is less than or equal to 0.1
Then STOP is confirmed.

Rule grid_7_true (#82)
If there is evidence of grid_6
And there is evidence of eval_region_grower
And there is evidence of execute_region_analysis
Then STOP is confirmed.
And (n*pug+pug+(1-pug)*0.4)/(n+1) is assigned to pug
And 1 is assigned to n

Rule grid_7_noncommittal (#83)
If there is evidence of grid_6
And there is no evidence of eval_region_grower
Then STOP is confirmed.
And 1 is assigned to n

Rule grid_7_false (#84)
If there is evidence of grid_6
And there is evidence of eval_region_grower
And there is no evidence of execute_region_analysis
Then STOP is confirmed.
And 1 is assigned to n

Rule terrain_1_true (#85)
If there is evidence of okay_terrain_effect
And there is evidence of eval_threshold
And percent_on-high_intensity_threshold is greater than or equal to 0
Then terrain_1 is confirmed.
And (n*pte+pte+(1-pte)*(0.6))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_1_false (#86)
If there is evidence of okay_terrain_effect
And there is evidence of eval_threshold
And percent_on-low_intensity_threshold is less than 0
Then terrain_1 is confirmed.
And (n*pte+pte+(pte)*((0-0.6)))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_2_true (#87)
If there is evidence of terrain_1
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is precisely equal to 1
Then terrain_2 is confirmed.

And (n*pte+pte+(1-pte)*(0.4))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_2_false (#88)
If there is evidence of terrain_1
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is not equal to 1
Then terrain_2 is confirmed.
And (n*pte+pte+(pte)*((0-0.4)))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_3_true (#89)
If there is evidence of terrain_2
And there is evidence of eval_edge_detector
And gradient-gradient_threshold is greater than or equal to 0
Then terrain_3 is confirmed.
And (n*pte+pte+(1-pte)*(0.4))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_3_false (#90)
If there is evidence of terrain_2
And there is evidence of eval_edge_detector
And gradient-gradient_threshold is less than 0
Then terrain_3 is confirmed.
And (n*pte+pte+(pte)*((0-0.4)))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_4_true (#91)
If there is evidence of terrain_3
And there is evidence of eval_local_contrast
And contrast.sigma-sigma_threshold is greater than or equal to 0
Then terrain_4 is confirmed.
And (n*pte+pte+(1-pte)*(0.2))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_4_false (#92)
If there is evidence of terrain_3
And there is evidence of eval_local_contrast
And contrast.sigma-sigma_threshold is less than 0
Then terrain_4 is confirmed.
And (n*pte+pte+(pte)*((0-0.2)))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_5_true (#93)
If there is evidence of terrain_4
And there is evidence of eval_bimodal_histogram
And there is evidence of bimodal_histogram
Then terrain_5 is confirmed.
And (n*pte+pte+(1-pte)*0.2)/(n+1) is assigned to pte
And n+1 is assigned to n

Rule terrain_5_false (#94)
If there is evidence of terrain_4

And there is evidence of eval_bimodal_histogram
And there is no evidence of bimodal_histogram
Then terrain_5 is confirmed.
And (n*pte+pte+(pte)*((0-0.2)))/(n+1) is assigned to pte
And n+1 is assigned to n

Rule trans_1_true (#95)
If there is evidence of okay_transmission_line
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is precisely equal to 1
Then trans_1 is confirmed.
And (n*ptl+ptl+(1-ptl)*(0.4))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule trans_1_false (#96)
If there is evidence of okay_transmission_line
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is not equal to 1
Then trans_1 is confirmed.
And (n*ptl+ptl+(ptl)*((0-0.4)))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule trans_2_true (#97)
If there is evidence of trans_1
And there is evidence of eval_periodic_highlights
And there is evidence of periodic_highlights
Then trans_2 is confirmed.
And (n*ptl+ptl+(1-ptl)*(0.4))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule trans_2_false (#98)
If there is evidence of trans_1
And there is evidence of eval_periodic_highlights
And there is no evidence of periodic_highlights
Then trans_2 is confirmed.
And (n*ptl+ptl+ptl*((0-0.4)))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule trans_3_true (#99)
If there is evidence of trans_2
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is less than 0
Then trans_3 is confirmed.
And (n*ptl+ptl+(1-ptl)*(0.4))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule trans_3_false (#100)
If there is evidence of trans_2
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And lines.angular_std_dev-curv_threshold is greater than or equal to 0

Then trans_3 is confirmed.
    And n+1 is assigned to n

Rule trans_4_true (#101)
If there is evidence of trans_3
    And there is evidence of eval_local_contrast
    And contrast.sigma-sigma_threshold is greater than or equal to 0
Then trans_4 is confirmed.
    And (n*ptl+ptl+(1-ptl)*0.4)/(n+1) is assigned to ptl
    And n+1 is assigned to n

Rule trans_4_false (#102)
If there is evidence of trans_3
    And there is evidence of eval_local_contrast
    And contrast.sigma is less than 2
Then trans_4 is confirmed.
    And (n*ptl+ptl+ptl*((0-0.4)))/(n+1) is assigned to ptl
    And n+1 is assigned to n

Rule trans_5_true (#103)
If there is evidence of trans_4
    And there is evidence of eval_bimodal_histogram
    And there is evidence of bimodal_histogram
Then trans_5 is confirmed.
    And (n*ptl+ptl+(1-ptl)*0.2)/(n+1) is assigned to ptl
    And n+1 is assigned to n

Rule trans_5_false (#104)
If there is evidence of trans_4
    And there is evidence of eval_bimodal_histogram
    And there is no evidence of bimodal_histogram
Then trans_5 is confirmed.
    And (n*ptl+ptl+ptl*((0-0.2)))/(n+1) is assigned to ptl
    And n+1 is assigned to n

Rule trans_6_true (#105)
If there is evidence of trans_5
    And there is evidence of eval_local_contrast
    And there is evidence of eval_DEM_drain_analysis
    And there is no evidence of motion
Then trans_6 is confirmed.
    And (n*ptl+ptl+(1-ptl)*0.4)/(n+1) is assigned to ptl
    And n+1 is assigned to n

Rule trans_6_noncommittal (#106)
If there is evidence of trans_5
    And there is evidence of eval_local_contrast
    And there is no evidence of eval_DEM_drain_analysis
Then trans_6 is confirmed.
    And n+1 is assigned to n

Rule trans_6_false (#107)
If there is evidence of trans_5
    And there is evidence of eval_local_contrast

And there is evidence of eval_DEM_drain_analysis
And there is evidence of motion
Then trans_6 is confirmed.
And (n*ptl+ptl+ptl*((0-0.4)))/(n+1) is assigned to ptl
And n+1 is assigned to n

Rule unimproved_1_true (#108)
If there is evidence of okay_unimproved_road
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is precisely equal to 1
Then unimproved_1 is confirmed.
And (n*pur+pur+(1-pur)*(0.2))/(n+1) is assigned to pur
And n+1 is assigned to n

Rule unimproved_1_false (#109)
If there is evidence of okay_unimproved_road
And there is evidence of eval_streak_detector
And there is evidence of eval_local_tangent
And ROUND(lines.number) is not equal to 1
Then unimproved_1 is confirmed.
And (n*pur+pur+pur*((0-1.0)))/(n+1) is assigned to pur
And n+1 is assigned to n

Rule unimproved_2_true (#110)
If there is evidence of unimproved_1
And there is evidence of eval_Mueller
And there is evidence of eval_threshold
And surface.roughness is "rough"
Then unimproved_2 is confirmed.
And (n*pur+pur+(1-pur)*0.4)/(n+1) is assigned to pur
And n+1 is assigned to n

Rule unimproved_2_noncommittal (#111)
If there is evidence of unimproved_1
And there is no evidence of eval_Mueller
Then unimproved_2 is confirmed.
And n+1 is assigned to n

Rule unimproved_2_false (#112)
If there is evidence of unimproved_1
And there is evidence of eval_Mueller
And there is evidence of eval_threshold
And surface.roughness is not "rough"
Then unimproved_2 is confirmed.
And n+1 is assigned to n

Rule unimproved_3_true (#113)
If there is evidence of unimproved_2
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-2*lane.size is less than or equal to 0
Then unimproved_3 is confirmed.
And (n*pur+pur+(1-pur)*0.4)/(n+1) is assigned to pur

And n+1 is assigned to n

Rule unimproved_3_false (#114)
If there is evidence of unimproved_2
And there is evidence of eval_local_tangent
And there is evidence of eval_width_profiler
And width-2*lane.size is greater than 0
Then unimproved_3 is confirmed.
And (n*pur+pur+pur*((0-0.4)))/(n+1) is assigned to pur
And n+1 is assigned to n

Rule unimproved_4_true (#115)
If there is evidence of unimproved_3
And there is evidence of eval_threshold
And there is evidence of eval_Mueller
And percent_even is greater than or equal to 50
Then unimproved_4 is confirmed.
And (n*pur+pur+(1-pur)*0.4)/(n+1) is assigned to pur
And n+1 is assigned to n

Rule unimproved_4_noncommittal (#116)
If there is evidence of unimproved_3
And there is evidence of eval_threshold
And there is no evidence of eval_Mueller
Then unimproved_4 is confirmed.
And n+1 is assigned to n

Rule unimproved_4_false (#117)
If there is evidence of unimproved_3
And there is evidence of eval_threshold
And there is evidence of eval_Mueller
And percent_even is less than 50
Then unimproved_4 is confirmed.
And n+1 is assigned to n

Rule unimproved_5_true (#118)
If there is evidence of unimproved_4
And there is evidence of eval_bimodal_histogram
And there is evidence of bimodal_histogram
Then unimproved_5 is confirmed.
And n+1 is assigned to n

Rule unimproved_5_false (#119)
If there is evidence of unimproved_4
And there is evidence of eval_bimodal_histogram
And there is no evidence of bimodal_histogram
Then unimproved_5 is confirmed.
And (n*pur+pur+(1-pur)*0.2)/(n+1) is assigned to pur
And n+1 is assigned to n

Appendix B   Rule Terminology Glossary

bimodal_histogram - Boolean value indicating whether there is a bimodal histogram in the local region of the candidate feature.

contrast.sigma - The average number of standard deviations above or below the local mean for pixels along the candidate feature.

crop_1 - Boolean indicating the crop_1 rule has been instantiated.

crop_2 - Boolean indicating the crop_2 rule has been instantiated.

crop_3 - Boolean indicating the crop_3 rule has been instantiated.

crop_4 - Boolean indicating the crop_4 rule has been instantiated.

crop_5 - Boolean indicating the crop_5 rule has been instantiated.

crop_6 - Boolean indicating the crop_6 rule has been instantiated.

curv_threshold - The threshold value set by the user that determines whether a candidate linear feature is classified as linear or curvilinear.

divided_1 - Boolean indicating the divided_1 rule has been instantiated.

divided_2 - Boolean indicating the divided_2 rule has been instantiated.

divided_3 - Boolean indicating the divided_3 rule has been instantiated.

divided_4 - Boolean indicating the divided_4 rule has been instantiated.

divided_5 - Boolean indicating the divided_5 rule has been instantiated.

divided_6 - Boolean indicating the divided_6 rule has been instantiated.

divided_7 - Boolean indicating the divided_7 rule has been instantiated.

eval_bimodal_histogram - Boolean that is set to indicate to the control executive that the test for a bimodal histogram is to be executed for the candidate feature.

eval_DEM_drain_analysis - Boolean that is set to indicate to the control executive that the module that compares the locations of DEM drain lines to those of the candidate feature is to be executed.

eval_edge_detector - Boolean that is set to indicate to the control executive that the edge detection module is to be executed for the candidate feature.

eval_local_contrast - Boolean that is set to indicate to the control executive that local regional statistics are to be compiled for the candidate feature.

eval_local_statistics - Boolean that is set to indicate to the control executive that local lineal statistics are to be compiled for the candidate linear feature.

eval_local_tangent - Boolean that is set to indicate to the control executive that the local tangent module is to be executed for the candidate feature.

eval_Mueller - Boolean that is set to indicate to the control executive that the polarimetric evaluation module is to be executed for the candidate feature.

eval_periodic_highlights - Boolean that is set to indicate to the control executive that the test for periodic highlights is to be executed for the candidate feature.

eval_Rayleigh_histogram - Boolean that is set to indicate to the control executive that the test for a Rayleigh histogram is to be executed for the candidate feature.

eval_region_grower - Boolean that is set to indicate to the control executive that the region growing module is to be executed for the candidate feature.

eval_streak_detector - Boolean that is set to indicate to the control executive that the streak detection module is to be executed for the candidate feature.

eval_threshold - Boolean that is set to indicate to the control executive that a simple threshold is to be executed for the candidate feature.

eval_width_profiler - Boolean that is set to indicate to the control executive that the width profiler module is to be executed for the candidate feature.

execute_region_analysis - Boolean that is set to indicate to the control executive that the region growing analysis module is to be executed for the candidate feature.

gradient - A floating point number indicating the average edge gradient for the pixels in the candidate feature.

gradient_threshold - The mean + 1 standard deviation of the global edge gradient for an image.

grid_1 - Boolean indicating the grid_1 rule has been instantiated.

grid_2 - Boolean indicating the grid_2 rule has been instantiated.

grid_3 - Boolean indicating the grid_3 rule has been instantiated.

grid_4 - Boolean indicating the grid_4 rule has been instantiated.

grid_5 - Boolean indicating the grid_5 rule has been instantiated.

grid_6 - Boolean indicating the grid_6 rule has been instantiated.

high_intensity_threshold - The intensity threshold set by the user that determines if a pixel will be turned on during a simple thresholding operation.

intensity.sigma - The standard deviation of intensity of pixels along a candidate feature as determined by the local lineal statistics module.

lane.size - The size in pixels of a nominal single lane road. This parameter may be determined from the SAR resolution and pixel size.

line_spacing_variance_threshold - The threshold set by the user that constrains the allowable variance in spacing of parallel lines detected in an image such that those lines may be classified as a divided highway.

lines.angular_std_dev - The standard deviation of the curvature of a linear feature as determined by the local tangent module.

lines.number - The number of parallel lines in the vicinity of the candidate feature as determined by a combination of the local tangent module and the threshold module.

lines.spacing_variance - The variance observed in the spacing of parallel lines in the vicinity of the candidate feature as determined by a combination of the local tangent module and the threshold module.

low_intensity_threshold - The intensity threshold set by the user that determines if a pixel will be turned on during a simple thresholding operation.

low_sigma_threshold - The user defined threshold for standard deviation of intensity in the local lineal statistics module. This parameter is used in relation to the fact that a small variation in the reflected power from a linear feature all along its length may indicate that the feature has a consistent dielectric constant. Moreover, the feature likely man-made.

motion - Boolean value set by the DEM drain line analysis module indicating the the candidate feature lies close to and parallel to a DEM drain line.

n - The number of instantiations of rules in a particular class of features.

okay_crop_boundary - The Boolean value set by the inference engine when it determines that it is okay to evaluate rules relating to crop boundaries.

okay_divided_highway - The Boolean value set by the inference engine when it determines that it is okay to evaluate rules relating to divided highways.

okay_terrain_effect - The Boolean value set by the inference engine determines when it determines that it is okay to evaluate rules relating to terrain effects.

okay_transmission_line - The Boolean value set by the inference engine determines when it determines that it is okay to evaluate rules relating to transmission lines.

okay_unimproved_road - The Boolean value set by the inference engine determines when it determines that it is okay to evaluate rules relating to unimproved roads.

okay_urban_grid - The Boolean value set by the inference engine determines when it determines that it is okay to evaluate rules relating to urban grids.

pcb - The certainty factor for crop boundaries.

pdh - The certainty factor for divided highways.

percent_even - The floating point number returned from the polarimetric analysis modules concerning the percentage of the pixels in the candidate feature that are consistent with an even number of bounces reflection.

percent_odd - The floating point number returned from the polarimetric analysis modules concerning the percentage of the pixels in the candidate feature that are consistent with an odd number of bounces reflection.

percent_on - The floating point number returned from the simple threshold module indicating the number of pixels in the candidate feature that fall outside of the **low_intensity_threshold** and the **high_intensity_threshold.**

percent_water - The floating point number indicating the water content of the region surrounding the candidate feature as derived from the polarimetric analysis module.

periodic_highlights - The Boolean value returned from the periodic highlights module indicating if the candidate feature is characterized as such.

pte - The certainty factor for terrain effects.

ptl - The certainty factor for transmission lines.

pug - The certainty factor for urban grids.

pur - The certainty factor for unimproved roads.

Rayleigh_histogram - The Boolean value indicating whether the intensity distribution in the region surrounding the candidate feature should be classified as Rayleigh. **Rayleigh_histogram** and **bimodal_histogram** always have opposite values.

sigma_threshold - The user defined threshold for segmenting outlyer in the local regional statistics module.

start - The symbolic link to the starting end of the reasoning chain. Suggesting **start** will initiate forward chaining.

stop - The symbolic link to the end of the reasoning chain. Suggesting **stop** will initiate backward chaining.

surface.roughness - A qualitative descriptor of the surface roughness of the candidate feature based upon the classification of its surface content.

terrain_1 - Boolean indicating the terrain_1 rule has been instantiated.

terrain_2 - Boolean indicating the terrain_2 rule has been instantiated.

terrain_3 - Boolean indicating the terrain_3 rule has been instantiated.

terrain_4 - Boolean indicating the terrain_4 rule has been instantiated.

terrain_5 - Boolean indicating the terrain_5 rule has been instantiated.

trans_1 - Boolean indicating the trans_1 rule has been instantiated.

trans_2 - Boolean indicating the trans_2 rule has been instantiated.

trans_3 - Boolean indicating the trans_3 rule has been instantiated.

trans_4 - Boolean indicating the trans_4 rule has been instantiated.

trans_5 - Boolean indicating the trans_5 rule has been instantiated.

trans_6 - Boolean indicating the trans_6 rule has been instantiated.

unimproved_1 - Boolean indicating the unimproved_1 rule has been instantiated.

unimproved_2 - Boolean indicating the unimproved_2 rule has been instantiated.

unimproved_3 - Boolean indicating the unimproved_3 rule has been instantiated.

unimproved_4 - Boolean indicating the unimproved_4 rule has been instantiated.

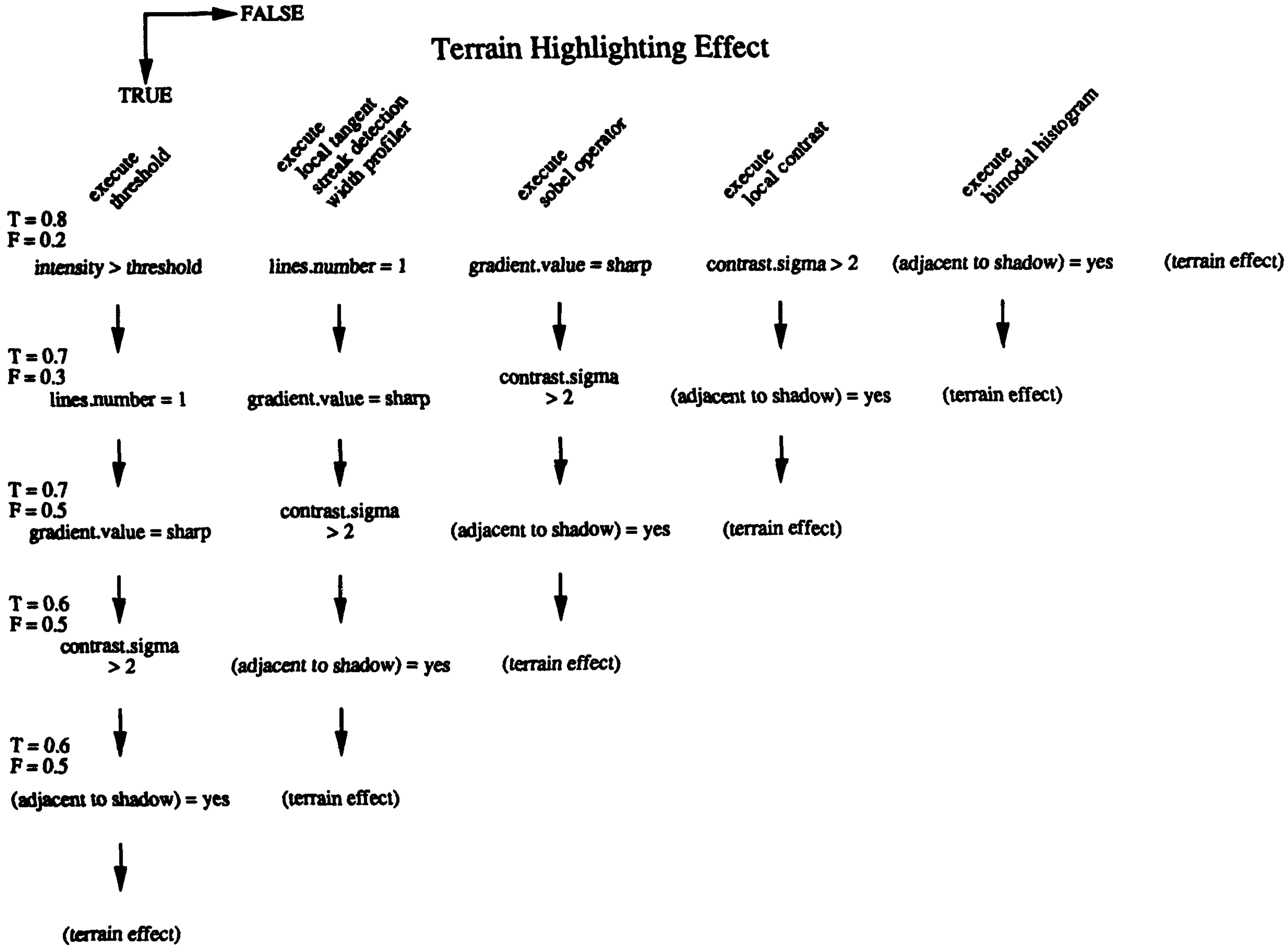
unimproved_5 - Boolean indicating the unimproved_5 rule has been instantiated.

width - The mean width in pixels of the candidate linear feature as determined by a combination of the local tangent module and the threshold module.

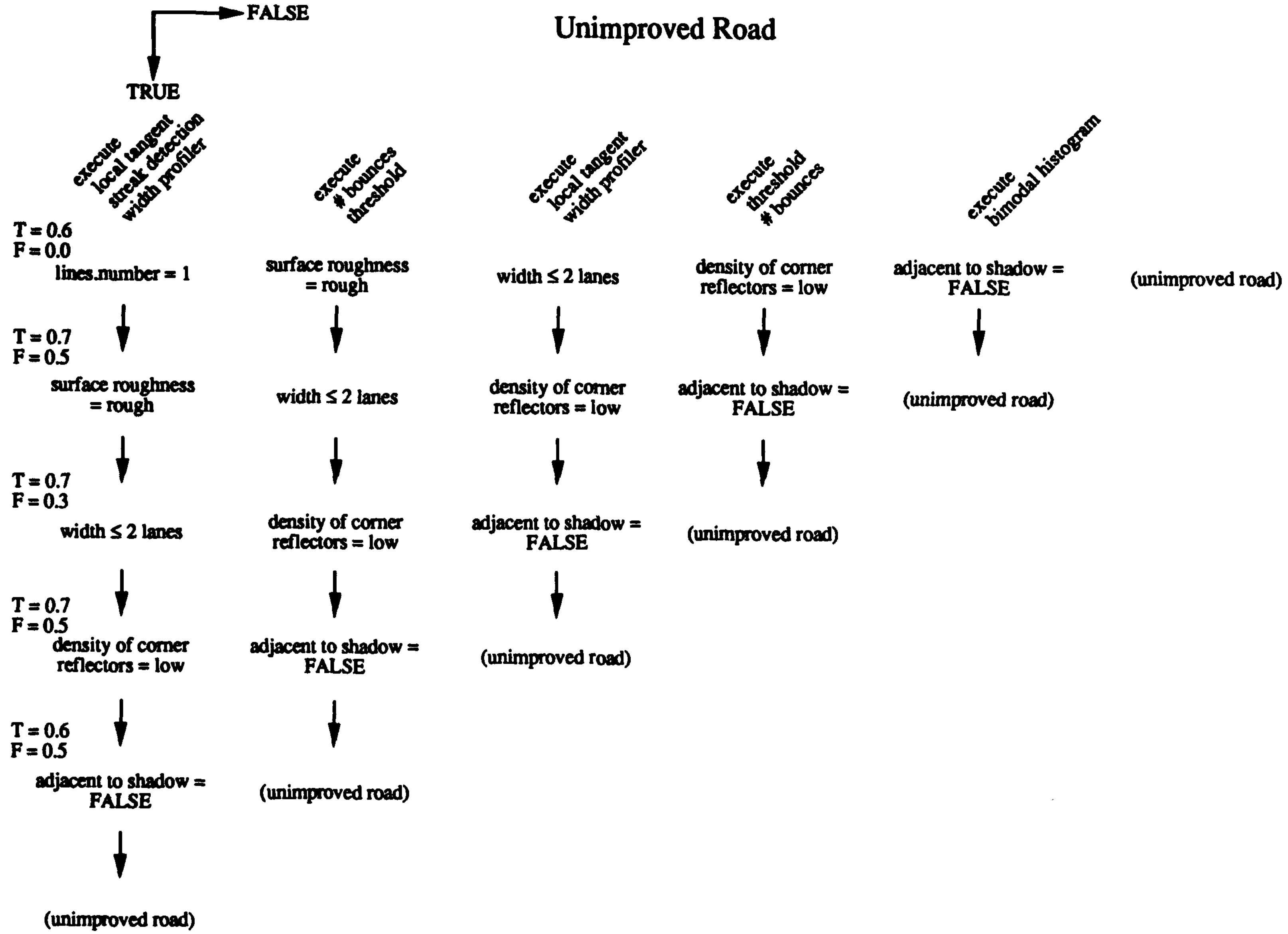
Appendix C Decision Trees

The decision trees presented in this section indicate the logical reasoning path for each of the five features described. In each decision tree, the certainty factors for TRUE and FALSE are given in the left column, and the required image processing subsystems are shown across the top row. The image processing modules apply only to the decision directly under them (which is repeated across the diagonal along the decision tree).

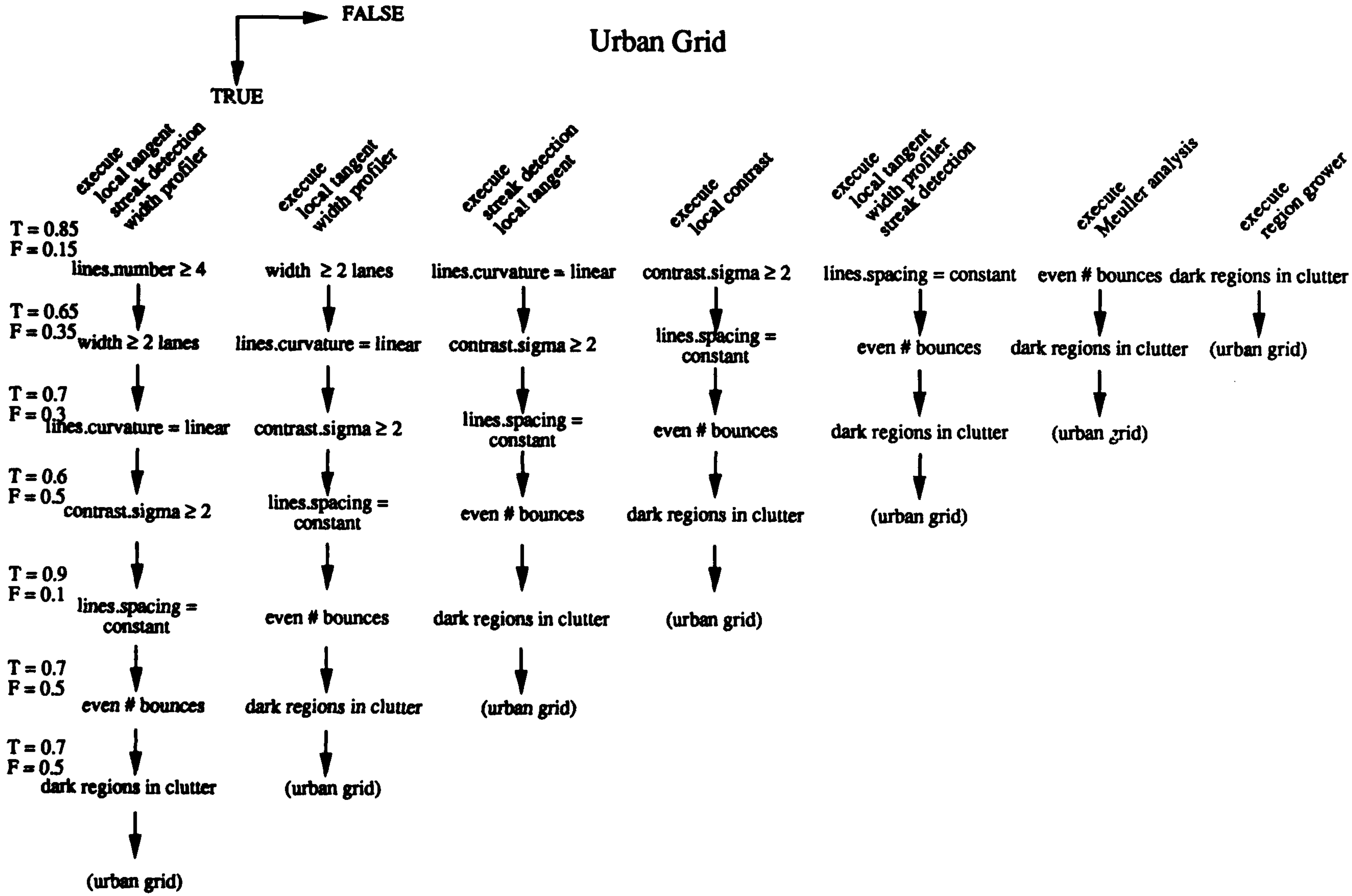
Terrain Highlighting Effect



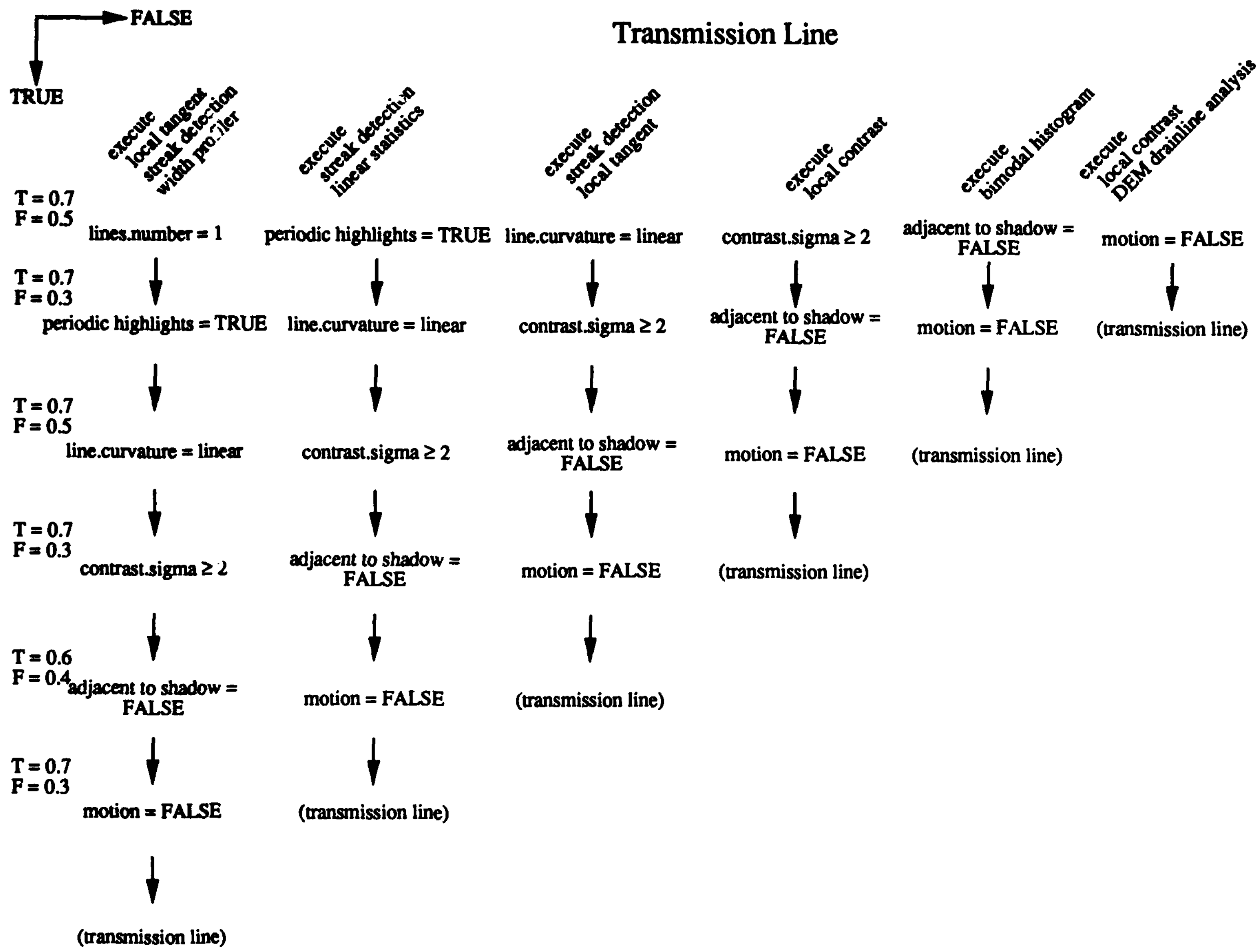
Unimproved Road



Urban Grid


FALSE
TRUE
execute local tangent streak detection width profiler
execute local tangent width profiler
execute streak detection local tangent
execute local contrast
execute local tangent width profiler streak detection
execute Meuller analysis
execute region grower
T = 0.85
F = 0.15
lines.number ≥ 4
T = 0.65
F = 0.35
width ≥ 2 lanes
T = 0.7
F = 0.3
lines.curvature = linear
T = 0.6
F = 0.5
contrast.sigma ≥ 2
T = 0.9
F = 0.1
lines.spacing = constant
T = 0.7
F = 0.5
even # bounces
T = 0.7
F = 0.5
dark regions in clutter
(urban grid)
width ≥ 2 lanes
lines.curvature = linear
contrast.sigma ≥ 2
lines.spacing = constant
even # bounces
dark regions in clutter
(urban grid)
lines.curvature = linear
contrast.sigma ≥ 2
lines.spacing = constant
even # bounces
dark regions in clutter
(urban grid)
contrast.sigma ≥ 2
lines.spacing = constant
even # bounces
dark regions in clutter
(urban grid)
lines.spacing = constant
even # bounces
dark regions in clutter
(urban grid)
even # bounces
dark regions in clutter
(urban grid)
dark regions in clutter
(urban grid)

Transmission Line



Crop Boundary